

# Actor-Critic Algorithms for Risk-Sensitive Reinforcement Learning

Prashanth L A <sup>\*1</sup> and Mohammad Ghavamzadeh <sup>†2</sup>

<sup>1</sup>INRIA Lille - Nord Europe, Team SequeL, FRANCE.

<sup>2</sup>Adobe Research & INRIA Lille - Team SequeL<sup>‡</sup>

## Abstract

In many sequential decision-making problems we may want to manage risk by minimizing some measure of variability in rewards in addition to maximizing a standard criterion. Variance related risk measures are among the most common risk-sensitive criteria in finance and operations research. However, optimizing many such criteria is known to be a hard problem. In this paper, we consider both discounted and average reward Markov decision processes. For each formulation, we first define a measure of variability for a policy, which in turn gives us a set of risk-sensitive criteria to optimize. For each of these criteria, we derive a formula for computing its gradient. We then devise actor-critic algorithms that operate on three timescales - a TD critic on the fastest timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we point out the difficulty in estimating the gradient of the variance of the return and incorporate simultaneous perturbation approaches to alleviate this. The average setting, on the other hand, allows for an actor update using compatible features to estimate the gradient of the variance. We establish the convergence of our algorithms to locally risk-sensitive optimal policies. Finally, we demonstrate the usefulness of our algorithms in a traffic signal control application.

**Keywords:** Markov decision process (MDP), reinforcement learning (RL), risk sensitive RL, actor-critic algorithms, multi-time-scale stochastic approximation, simultaneous perturbation stochastic approximation (SPSA), smoothed functional (SF).

## 1 Introduction

The usual optimization criteria for an infinite horizon Markov decision process (MDP) are the *expected sum of discounted rewards* and the *average reward* [35, 3]. Many algorithms have been developed to maximize these criteria both when the model of the system is known (planning) and unknown (learning) [5, 44]. These algorithms can be categorized to **value function-based** methods that are mainly based on the two celebrated dynamic programming algorithms *value iteration* and *policy iteration*; and **policy gradient** methods that are based on updating the policy parameters in the direction of the gradient of a performance measure, i.e., the value function of the initial state or the average reward. Policy gradient methods estimate the gradient of the performance measure either without using an explicit representation of the value function (e.g., [50, 28, 2]) or using such a representation in which case they are referred to as *actor-critic* algorithms (e.g., [45, 25, 31, 10, 11]). Using an explicit representation for value function (e.g., linear function approximation) by actor-critic algorithms reduces the variance of the gradient estimate with the cost of adding it a bias.

Actor-critic methods were among the earliest to be investigated in RL [1, 42]. They comprise a family of reinforcement learning (RL) methods that maintain two distinct algorithmic components: An *Actor*, whose role is to maintain and update an action-selection policy; and a *Critic*, whose role is to estimate the value function

<sup>\*</sup>prashanth.la@inria.fr

<sup>†</sup>mohammad.ghavamzadeh@inria.fr

<sup>‡</sup>currently at Adobe Research, on leave of absence from INRIA.

associated with the actor’s policy. Thus, the critic addresses a problem of *prediction*, whereas the actor is concerned with *control*. A common practice is to update the policy parameters using stochastic gradient ascent, and to estimate the value-function using some form of temporal difference (TD) learning [43].

However in many applications, we may prefer to minimize some measure of *risk* as well as maximizing a usual optimization criterion. In such cases, we would like to use a criterion that incorporates a penalty for the *variability* induced by a given policy. This variability can be due to two types of uncertainties: **1)** uncertainties in the model parameters, which is the topic of *robust* MDPs (e.g., [30, 20, 51]), and **2)** the inherent uncertainty related to the stochastic nature of the system, which is the topic of *risk-sensitive* MDPs (e.g., [23, 37, 21]).

In risk-sensitive sequential decision-making, the objective is to maximize a risk-sensitive criterion such as the expected exponential utility [23], a variance related measure [37, 21], or the percentile performance [22]. Unfortunately, when we include a measure of risk in our optimality criteria, the corresponding optimal policy is usually no longer Markovian stationary (e.g., [21]) and/or computing it is not tractable (e.g., [21, 27]). Although risk-sensitive sequential decision-making has a long history in operations research and finance, it has only recently grabbed attention in the machine learning community. Most of the work on this topic (including those mentioned above) has been in the context of MDPs (when the model of the system is known) and much less work has been done within the reinforcement learning (RL) framework (when the model is unknown and all the information about the system is obtained from the samples resulted from the agent’s interaction with the environment). In risk-sensitive RL, we can mention the work by Borkar [14, 15, 18] who considered the expected exponential utility and the one by Tamar et al. [47] on several variance related measures. Tamar et al. [47] study stochastic shortest path problems, and in this context, propose a policy gradient algorithm (and in a more recent work [46] an actor-critic algorithm) for maximizing several risk-sensitive criteria that involve both the expectation and variance of the *return* random variable (defined as the sum of the rewards that the agent obtains in an episode).

In this paper,<sup>1</sup> we develop actor-critic algorithms for optimizing variance-related risk measures in both discounted and average reward MDPs. Our contributions can be summarized as follows:

- In the discounted reward setting we define the measure of variability as the *variance of the return* (similar to [47]). We formulate a constrained optimization problem with the aim of maximizing the mean of the return subject to its variance being bounded from above. We employ the Lagrangian relaxation procedure [4] and derive a formula for the gradient of the Lagrangian. Since this requires the gradient of the value function at every state of the MDP (see the discussion in Sections 3 and 4), we estimate the gradient of the Lagrangian using two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) [38] and *smoothed functional* (SF) [24], resulting in two separate discounted reward actor-critic algorithms. In addition, we also propose second-order algorithms with a Newton step, using both SPSA and SF.
- In the average reward formulation, we first define the measure of variability as the *long-run variance* of a policy, and using a constrained optimization problem similar to the discounted case, derive an expression for the gradient of the Lagrangian. We then develop an actor-critic algorithm with *compatible features* [45, 31] to estimate the gradient and to optimize the policy parameters.
- Using the ordinary differential equations (ODE) approach, we establish the asymptotic convergence of our algorithms to locally risk-sensitive optimal policies.
- We demonstrate the usefulness of our discounted and average reward risk-sensitive actor-critic algorithms in a traffic signal control application.

It is important to note that our both discounted and average reward algorithms can be easily extended to other variance related risk criteria such as the Sharpe ratio, which is popular in financial decision-making [36] (see Sections 4.6 and 6.2 for more details).

In comparison to [47] and [46], which are the most closely related contributions, we would like to point out the following:

- (i) The authors develop policy gradient and actor-critic methods for stochastic shortest path problems in [47] and

<sup>1</sup>This paper is an extension of an earlier work by the authors [34] and includes novel second order methods in the discounted setting, detailed proofs of all proposed algorithms, and additional experimental results.

[46], respectively. On the other hand, we devise actor-critic algorithms for both discounted and average reward MDP settings.; and

(ii) More importantly, we note the difficulty in the discounted formulation that requires to estimate the gradient of the value function at every state of the MDP and also sample from two different distributions. This precludes us from using *compatible features* - a method that has been employed successfully in actor-critic algorithms in a risk-neutral setting (cf. [11]) as well as more recently in [46] for a risk-sensitive stochastic shortest path setting. We alleviate the above mentioned problems for the discounted setting by employing simultaneous perturbation based schemes for estimating the gradient in the first order methods and Hessian in the second order methods, that we propose.

The rest of the paper is organized as follows: In Section 2, we describe the RL setting. In Section 3, we describe the risk-sensitive MDP in the discounted setting and propose actor-critic algorithms for this setting in Section 4. In Section 5, we present the risk measure for the average setting and propose an actor-critic algorithm that optimizes this risk measure in Section 6. In Section 7, we describe the experimental setup and present the results in both average and discounted cost settings. Finally, in Section 8, we provide the concluding remarks and outline a few future research directions.

## 2 Preliminaries

We consider sequential decision-making tasks that can be formulated as a reinforcement learning (RL) problem. In RL, an agent interacts with a dynamic, stochastic, and incompletely known environment, with the goal of optimizing some measure of its *long-term* performance. This interaction is often modeled as a Markov decision process (MDP). A MDP is a tuple  $(\mathcal{X}, \mathcal{A}, R, P, x^0)$  where  $\mathcal{X} = \{1, \dots, n\}$  and  $\mathcal{A} = \{1, \dots, m\}$  are the state and action spaces;  $R(x, a), x \in \mathcal{X}, a \in \mathcal{A}$  is the reward random variable whose expectation is denoted by  $r(x, a) = \mathbb{E}[R(x, a)]$ ;  $P(\cdot|x, a)$  is the transition probability distribution; and  $x^0 \in \mathcal{X}$  is the initial state<sup>2</sup>.

The rule according to which the agent acts in its environment (selects action at each state) is called a *policy*. A Markovian stationary policy  $\mu(\cdot|x)$  is a probability distribution over actions, conditioned on the current state  $x$ . The goal in a RL problem is to find a policy that optimizes the long-term performance measure of interest, e.g., maximizes the *expected discounted sum of rewards* or the *average reward*.

In policy gradient and actor-critic methods, we define a class of parameterized stochastic policies  $\{\mu(\cdot|x; \theta), x \in \mathcal{X}, \theta \in \Theta \subseteq \mathbb{R}^{\kappa_1}\}$ , estimate the gradient of the performance measure w.r.t. the policy parameters  $\theta$  from the observed system trajectories, and then improve the policy by adjusting its parameters in the direction of the gradient. Since in this setting a policy  $\mu$  is represented by its  $\kappa_1$ -dimensional parameter vector  $\theta$ , policy dependent functions can be written as a function of  $\theta$  in place of  $\mu$ . So, we use  $\mu$  and  $\theta$  interchangeably in the paper.

We make the following assumptions on the policy, parameterized by  $\theta$ :

**(A1)** For any state-action pair  $(x, a) \in \mathcal{X} \times \mathcal{A}$ , the policy  $\mu(a|x; \theta)$  is continuously differentiable in the parameter  $\theta$ .

**(A2)** The Markov chain induced by any policy  $\theta$  is irreducible and aperiodic.

The above assumptions are standard requirements in policy gradient and actor-critic methods.

Finally, we denote by  $d^\mu(x)$  and  $\pi^\mu(x, a) = d^\mu(x)\mu(a|x)$ , the stationary distribution of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$ , respectively. Similarly in the discounted formulation, we define the  $\gamma$ -discounted visiting distribution of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$  as  $d_\gamma^\mu(x|x^0) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(x_t = x|x_0 = x^0; \mu)$  and  $\pi_\gamma^\mu(x, a|x^0) = d_\gamma^\mu(x|x^0)\mu(a|x)$ .

<sup>2</sup>Our algorithms can be easily extended to a setting where the initial state is determined by a distribution.

### 3 Discounted Reward Setting

For a given policy  $\mu$ , we define the return of a state  $x$  (state-action pair  $(x, a)$ ) as the sum of discounted rewards encountered by the agent when it starts at state  $x$  (state-action pair  $(x, a)$ ) and then follows policy  $\mu$ , i.e.,

$$D^\mu(x) = \sum_{t=0}^{\infty} \gamma^t R(x_t, a_t) \mid x_0 = x, \mu,$$

$$D^\mu(x, a) = \sum_{t=0}^{\infty} \gamma^t R(x_t, a_t) \mid x_0 = x, a_0 = a, \mu.$$

The expected value of these two random variables are the value and action-value functions of policy  $\mu$ , i.e.,

$$V^\mu(x) = \mathbb{E}[D^\mu(x)] \quad \text{and} \quad Q^\mu(x, a) = \mathbb{E}[D^\mu(x, a)].$$

The goal in the standard (risk-neutral) discounted reward formulation is to find an optimal policy  $\mu^* = \arg \max_{\mu} V^\mu(x^0)$ , where  $x^0$  is the initial state of the system.

The most common measure of the *variability* in the stream of rewards is the *variance of the return*, defined by

$$\Lambda^\mu(x) \triangleq \mathbb{E}[D^\mu(x)^2] - V^\mu(x)^2 = U^\mu(x) - V^\mu(x)^2. \quad (1)$$

The above measure was first introduced by Sobel [37]. Note that

$$U^\mu(x) \triangleq \mathbb{E}[D^\mu(x)^2]$$

is the *square reward value function* of state  $x$  under policy  $\mu$ . On similar lines, we define the *square reward action-value function* of state-action pair  $(x, a)$  under policy  $\mu$  as

$$W^\mu(x, a) \triangleq \mathbb{E}[D^\mu(x, a)^2].$$

From the Bellman equation of  $\Lambda^\mu(x)$ , proposed by Sobel [37], it is straightforward to derive the following Bellman equations for  $U^\mu(x)$  and  $W^\mu(x, a)$ :

$$U^\mu(x) = \sum_a \mu(a|x) r(x, a)^2 + \gamma^2 \sum_{a, x'} \mu(a|x) P(x'|x, a) U^\mu(x') \\ + 2\gamma \sum_{a, x'} \mu(a|x) P(x'|x, a) r(x, a) V^\mu(x'), \quad (2)$$

$$W^\mu(x, a) = r(x, a)^2 + \gamma^2 \sum_{x'} P(x'|x, a) U^\mu(x') + 2\gamma r(x, a) \sum_{x'} P(x'|x, a) V^\mu(x').$$

Although  $\Lambda^\mu$  of (1) satisfies a Bellman equation, unfortunately, it lacks the monotonicity property of dynamic programming (DP), and thus, it is not clear how the related risk measures can be optimized by standard DP algorithms [37]. Policy gradient and actor-critic algorithms are good candidates to deal with this risk measure.

We consider the following risk-sensitive measure for discounted MDPs: for a given  $\alpha > 0$ ,

$$\max_{\theta} V^\theta(x^0) \quad \text{subject to} \quad \Lambda^\theta(x^0) \leq \alpha. \quad (3)$$

It is important to note that the algorithms proposed in this paper can be used for any risk-sensitive measure that is based on the variance of the return such as

1.  $\min_{\theta} \Lambda^\theta(x^0) \quad \text{subject to} \quad V^\theta(x^0) \geq \alpha,$
2.  $\max_{\theta} V^\theta(x^0) - \alpha \sqrt{\Lambda^\theta(x^0)},$

3. Maximizing the Sharpe Ratio, i.e.,  $\max_{\theta} V^{\theta}(x^0)/\sqrt{\Lambda^{\theta}(x^0)}$ . Sharpe Ratio (SR) is a popular risk measure in financial decision-making [36]. Section 4.6 presents extensions of our proposed discounted reward algorithms to optimize the Sharpe ration.

To solve (3), we employ the Lagrangian relaxation procedure [4] to convert it to the following unconstrained problem:

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq -V^{\theta}(x^0) + \lambda(\Lambda^{\theta}(x^0) - \alpha) \right), \quad (4)$$

where  $\lambda$  is the Lagrange multiplier. The goal here is to find the saddle point of  $L(\theta, \lambda)$ , i.e., a point  $(\theta^*, \lambda^*)$  that satisfies

$$L(\theta, \lambda^*) \geq L(\theta^*, \lambda^*) \geq L(\theta^*, \lambda), \forall \theta \in \Theta, \forall \lambda > 0.$$

This is achieved by descending in  $\theta$  and ascending in  $\lambda$  using the following gradients:

$$\nabla_{\theta} L(\theta, \lambda) = -\nabla_{\theta} V^{\theta}(x^0) + \lambda \nabla_{\theta} \Lambda^{\theta}(x^0) \quad \text{and} \quad \nabla_{\lambda} L(\theta, \lambda) = \Lambda^{\theta}(x^0) - \alpha.$$

Since  $\nabla_{\theta} \Lambda^{\theta}(x^0) = \nabla_{\theta} U^{\theta}(x^0) - 2V^{\theta}(x^0)\nabla_{\theta} V^{\theta}(x^0)$ , in order to compute  $\nabla_{\theta} \Lambda^{\theta}(x^0)$  it would be enough to calculate  $\nabla_{\theta} V^{\theta}(x^0)$  and  $\nabla_{\theta} U^{\theta}(x^0)$ . Using the above definitions, we are now ready to derive the expressions for the gradient of  $V^{\theta}(x^0)$  and  $U^{\theta}(x^0)$ , which in turn constitute the main ingredients in calculating  $\nabla_{\theta} L(\theta, \lambda)$ .

**Lemma 1.** *Under (A1) and (A2), we have*

$$\begin{aligned} (1 - \gamma)\nabla_{\theta} V^{\theta}(x^0) &= \sum_{x,a} \pi_{\gamma}^{\theta}(x, a|x^0) \nabla \log \mu(a|x; \theta) Q^{\theta}(x, a), \\ (1 - \gamma^2)\nabla_{\theta} U^{\theta}(x^0) &= \sum_{x,a} \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) \nabla \log \mu(a|x; \theta) W^{\theta}(x, a) \\ &\quad + 2\gamma \sum_{x,a,x'} \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) P(x'|x, a) r(x, a) \nabla_{\theta} V^{\theta}(x'), \end{aligned}$$

where  $\tilde{d}_{\gamma}^{\theta}(x|x^0)$  and  $\tilde{\pi}_{\gamma}^{\theta}(x, a|x^0)$  are the  $\gamma^2$ -discounted visiting distributions of state  $x$  and state-action pair  $(x, a)$  under policy  $\mu$ , respectively, and are defined as

$$\begin{aligned} \tilde{d}_{\gamma}^{\theta}(x|x^0) &= (1 - \gamma^2) \sum_{t=0}^{\infty} \gamma^{2t} \Pr(x_t = x | x_0 = x^0; \theta), \\ \tilde{\pi}_{\gamma}^{\theta}(x, a|x^0) &= \tilde{d}_{\gamma}^{\theta}(x|x^0) \mu(a|x). \end{aligned}$$

*Proof.* The proof of  $\nabla V^{\theta}(x^0)$  is standard and can be found, for instance, in [31]. To prove  $\nabla U^{\theta}(x^0)$ , we start by the fact that from (2) we have  $U(x) = \sum_a \mu(x|a) W(x, a)$ . If we take the derivative w.r.t.  $\theta$  from both sides of this equation and obtain

$$\begin{aligned}
\nabla U(x^0) &= \sum_a \nabla \mu(x^0|a) W(x^0, a) + \sum_a \mu(a|x^0) \nabla W(x^0, a) \\
&= \sum_a \nabla \mu(a|x^0) W(x^0, a) + \sum_a \mu(a|x^0) \nabla \left[ r(x^0, a)^2 + \gamma^2 \sum_{x'} P(x'|x^0, a) U(x') \right. \\
&\quad \left. + 2\gamma r(x^0, a) \sum_{x'} P(x'|x^0, a) V(x') \right] \\
&= \underbrace{\sum_a \nabla \mu(x^0|a) W(x^0, a) + 2\gamma \sum_{a, x'} \mu(a|x^0) r(x^0, a) P(x'|x^0, a) \nabla V(x')}_{h(x^0)} \\
&\quad + \gamma^2 \sum_{a, x'} \mu(a|x^0) P(x'|x^0, a) \nabla U(x') \\
&= h(x^0) + \gamma^2 \sum_{a, x'} \mu(a|x^0) P(x'|x^0, a) \nabla U(x') \\
&= h(x^0) + \gamma^2 \sum_{a, x'} \mu(a|x^0) P(x'|x^0, a) \nabla \left[ h(x') + \gamma^2 \sum_{a', x''} \mu(a'|x') P(x''|x', a') \nabla U(x'') \right].
\end{aligned} \tag{5}$$

By unrolling the last equation using the definition of  $\nabla U(x)$  from (5), we obtain

$$\begin{aligned}
\nabla U(x^0) &= \sum_{t=0}^{\infty} \gamma^{2t} \sum_x \Pr(x_t = x | x_0 = x^0) h(x) = \frac{1}{1 - \gamma^2} \sum_x \tilde{d}_\gamma(x|x^0) h(x) \\
&= \frac{1}{1 - \gamma^2} \left[ \sum_{x, a} \tilde{d}_\gamma(x|x^0) \mu(a|x) \nabla \log \mu(a|x) W(x, a) \right. \\
&\quad \left. + 2\gamma \sum_{x, a, x'} \tilde{d}_\gamma(x|x^0) \mu(a|x) r(x, a) P(x'|x, a) \nabla V(x') \right] \\
&= \frac{1}{1 - \gamma^2} \left[ \sum_{x, a} \tilde{\pi}_\gamma(x, a|x^0) \nabla \log \mu(a|x) W(x, a) \right. \\
&\quad \left. + 2\gamma \sum_{x, a, x'} \tilde{\pi}_\gamma(x, a|x^0) r(x, a) P(x'|x, a) \nabla V(x') \right]. \quad \blacksquare
\end{aligned}$$

□

It is challenging to devise an efficient method to estimate  $\nabla_\theta L(\theta, \lambda)$  using the gradient formulas of Lemma 1. This is mainly because

1. two different sampling distributions,  $\pi_\gamma^\theta$  and  $\tilde{\pi}_\gamma^\theta$ , are used for  $\nabla V^\theta(x^0)$  and  $\nabla U^\theta(x^0)$ , and
2.  $\nabla V^\theta(x')$  appears in the second sum of  $\nabla U^\theta(x^0)$  equation, which implies that we need to estimate the gradient of the value function  $V^\theta$  at every state of the MDP, and not just at the initial state  $x^0$ .

To alleviate the above mentioned problems, we borrow the principle of simultaneous perturbation for estimating the gradient  $\nabla_\theta L(\theta, \lambda)$  and develop novel risk-sensitive actor-critic algorithms in the following section.

## 4 Discounted Reward Risk-Sensitive Actor-Critic Algorithms

In this section, we present actor-critic algorithms for optimizing the risk-sensitive measure (3). These algorithms are based on two simultaneous perturbation methods: *simultaneous perturbation stochastic approximation* (SPSA) and *smoothed functional* (SF).

Simultaneous perturbation methods have been popular in the field of stochastic optimization and the reader is referred to [13] for a textbook introduction. First introduced in [38], the idea of SPSA is to perturb each coordinate of a parameter vector uniformly using Rademacher random variable, in the quest for finding the minimum of a function that is only observable via simulation. Traditional gradient schemes require  $2\kappa_1$  evaluations of the function, where  $\kappa_1$  is the parameter dimension. On the other hand, SPSA requires only two evaluations irrespective of the parameter dimension and hence is an efficient scheme, especially useful in high-dimensional settings. While a one-simulation variant of SPSA was proposed in [39], the original two-simulation SPSA algorithm is preferred as it is more efficient and also seen to work better than its one-simulation variant. Later enhancements to the original SPSA scheme include using deterministic perturbation using certain Hadamard matrices [9] and second-order methods that estimate Hessian using SPSA [40, 6].

The SF schemes are another class of simultaneous perturbation methods, which again perturb each coordinate of the parameter vector uniformly. However, unlike SPSA, Gaussian random variables are used here for the perturbation. Originally proposed in [24], the SF schemes have been studied and enhanced in later works such as [41, 7]. Further, [12] proposes both SPSA and SF like schemes for constrained optimization.

## 4.1 Algorithm Structure

For the purpose of finding an optimal risk-sensitive policy, a standard procedure would update the policy parameter  $\theta$  and Lagrange multiplier  $\lambda$  in two nested loops as follows:

- An inner loop that descends in  $\theta$  using the gradient of the Lagrangian  $L(\theta, \lambda)$  w.r.t.  $\theta$ , and
- An outer loop that ascends in  $\lambda$  using the gradient of the Lagrangian  $L(\theta, \lambda)$  w.r.t.  $\lambda$ .

Using multi-timescale stochastic approximation [17, Chapter 6], the two loops above can run in parallel, as follows:

$$\theta_{t+1} = \Gamma[\theta_t - \zeta_2(t)A_t^{-1}\nabla_{\theta}L(\theta, \lambda)], \quad (6)$$

$$\lambda_{t+1} = \Gamma_{\lambda}[\lambda_t + \zeta_1(t)\nabla_{\lambda}L(\theta, \lambda)], \quad (7)$$

where,  $A_t$  is a positive-definite matrix, chosen in an algorithm-specific manner and  $\Gamma, \Gamma_{\lambda}$  are certain projection operators that keep the iterates  $(\theta_t, \lambda_t)$  stable. Further,  $\zeta_1(t), \zeta_2(t)$  are step-sizes selected such that  $\theta$  update is on the faster and  $\lambda$  update is on the slower timescale. For the first order methods,  $A_t = I$  ( $I$  is the identity matrix), while for the second order methods  $A_t = \nabla_{\theta}^2 L(\theta_t, \lambda_t)$ . Note that another timescale  $\zeta_3(t)$  that is the fastest is used for the TD-critic, which provides the estimate of the Lagrangian for a given  $(\theta, \lambda)$ .

We make the following assumptions on the step-size schedules:

**(A3)** The step size schedules  $\{\zeta_3(t)\}$ ,  $\{\zeta_2(t)\}$ , and  $\{\zeta_1(t)\}$  satisfy

$$\sum_t \zeta_1(t) = \sum_t \zeta_2(t) = \sum_t \zeta_3(t) = \infty, \quad (8)$$

$$\sum_t \zeta_1(t)^2, \quad \sum_t \zeta_2(t)^2, \quad \sum_t \zeta_3(t)^2 < \infty, \quad (9)$$

$$\zeta_1(t) = o(\zeta_2(t)), \quad \zeta_2(t) = o(\zeta_3(t)). \quad (10)$$

Equations 8 and 9 are standard step-size conditions in stochastic approximation algorithms, and Equation 10 ensures that the critic updates are on the fastest time-scale  $\{\zeta_3(t)\}$ , the policy parameter update is on the intermediate time-scale  $\{\zeta_2(t)\}$ , and the Lagrange multiplier update is on the slowest time-scale  $\{\zeta_1(t)\}$ .

While  $\nabla_{\lambda}L(\theta, \lambda)$  has a particularly simple form of  $(\Lambda^{\theta}(x^0) - \alpha)$ , it is not possible to obtain closed form expressions for  $\nabla_{\theta}L(\theta, \lambda)$ . To alleviate this, we employ simultaneous perturbation schemes for estimating the gradient (and in the case of second order methods, the Hessian) of the Lagrangian  $L(\theta, \lambda)$ . The idea in these methods is to estimate the gradients  $\nabla_{\theta}V^{\theta}(x^0)$  and  $\nabla_{\theta}U^{\theta}(x^0)$  (needed for estimating the gradient  $\nabla_{\theta}L(\theta, \lambda)$ )

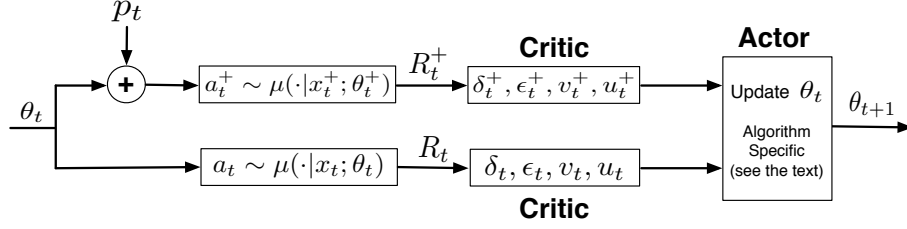


Figure 1: The overall flow of our simultaneous perturbation based actor-critic algorithms.

using two simulated trajectories of the system corresponding to policies with parameters  $\theta_t$  and  $\theta_t^+ = \theta_t + p_t$ . Here  $p_t$  is a perturbation vector that is specific to the algorithm.

Based on the order, our algorithms can be classified as:

1. **First order:** This corresponds to  $A_t = I$  in (6). The proposed algorithms here include RS-SPSA-G and RS-SF-G, where the former estimates the gradient using SPSA, while the latter uses SF. These algorithms use the following choice for the perturbation vector:  $p_t = \beta \Delta_t$ . Here  $\beta > 0$  is a positive constant and  $\Delta_t$  is a perturbation random variable, i.e., a  $\kappa_1$ -vector of independent Rademacher (for SPSA) and Gaussian  $\mathcal{N}(0, 1)$  (for SF) random variables.
2. **Second order:** This corresponds to  $A_t = \nabla^2 L(\theta_t, \lambda_t)$  in (6). The proposed algorithms here include RS-SPSA-N and RS-SF-N, where the former uses SPSA for gradient/Hessian estimates and the latter employs SF for the same. These algorithms use the following choice for perturbation vector: For RS-SPSA-N,  $p_t = \beta \Delta_t + \beta \hat{\Delta}_t$ ,  $\beta > 0$  is a positive constant and  $\Delta_t$  and  $\hat{\Delta}_t$  are perturbation parameters that are  $\kappa_1$ -vectors of independent Rademacher random variables, respectively. For RS-SF-N,  $p_t = \beta \Delta_t$ , where  $\Delta_t$  is a  $\kappa_1$  vector of Gaussian  $\mathcal{N}(0, 1)$  random variables.

---

**Algorithm 1** Template of the Risk-Sensitive Discounted Reward Actor-Critic Algorithms

---

**Input:** parameterized policy  $\mu(\cdot|\cdot; \theta)$  and value function feature vectors  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$   
**Initialization:** policy parameter  $\theta = \theta_0$ ; value function weight vectors  $v = v_0$  and  $v^+ = v_0^+$ ; square value function weight vectors  $u = u_0$  and  $u^+ = u_0^+$ ; initial state  $x_0 \sim P_0(x)$   
**for**  $t = 0, 1, 2, \dots$  **do**  
    Draw action  $a_t \sim \mu(\cdot|x_t; \theta_t)$ , observe next state  $x_{t+1}$  and reward  $R(x_t, a_t)$   
    Draw action  $a_t^+ \sim \mu(\cdot|x_t^+; \theta_t^+)$ , observe next state  $x_{t+1}^+$  and reward  $R(x_t^+, a_t^+)$   
    **Critic Update:** see (14) and (15) in the text  
    **Actor Update:** Algorithm-Specific  
    **Lagrange Multiplier Update:** see (20) in the text  
**end for**  
**return** policy and value function parameters  $\theta, \lambda, v, u$

---

The overall flow of our proposed actor-critic algorithms is illustrated in Figure 1 and Algorithm 1. The main steps involved at each time instant  $t$  are the following:

- 1) **Unperturbed Simulation:** Take action  $a_t \sim \mu(\cdot|x_t; \theta_t)$ , observe the reward  $R(x_t, a_t)$ , and the next state  $x_{t+1}$  in the first trajectory.
- 2) **Perturbed Simulation:** Take action  $a_t^+ \sim \mu(\cdot|x_t^+; \theta_t^+)$ , observe the reward  $R(x_t^+, a_t^+)$ , and the next state  $x_{t+1}^+$  in the second trajectory.
- 3) **Critic Update:** Using the method of temporal differences (TD) [42], estimate the value functions  $\hat{V}^{\theta_t}(x^0)$  and  $\hat{V}^{\theta_t^+}(x^0)$ , and square value functions  $\hat{U}^{\theta_t}(x^0)$  and  $\hat{U}^{\theta_t^+}(x^0)$ , corresponding to the policy parameter  $\theta_t$  and  $\theta_t^+$ .



**4) Actor Update:** Estimate the gradient/Hessian of  $\widehat{V}^\theta(x^0)$  and  $\widehat{U}^\theta(x^0)$ , and hence the gradient/Hessian of Lagrangian  $L(\theta, \lambda)$ , using either SPSA (16) or SF (17) methods. Using these estimates, update the policy parameter  $\theta$  in the descent direction using either a gradient or a Newton decrement, and the Lagrange multiplier  $\lambda$  in the ascent direction.

In the next section, we describe the TD-critic and subsequently, in Sections 4.3–4.4, present the first and second order actor critic algorithms, respectively.

## 4.2 TD-Critic

In our actor-critic algorithms, the critic uses linear approximation for the value and square value functions, i.e.,  $\widehat{V}(x) \approx v^\top \phi_v(x)$  and  $\widehat{U}(x) \approx u^\top \phi_u(x)$ , where the features  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$  are from low-dimensional spaces  $\mathbb{R}^{\kappa_2}$  and  $\mathbb{R}^{\kappa_3}$ , respectively. Let  $\Phi_v$  and  $\Phi_u$  denote  $n \times \kappa_2$  and  $n \times \kappa_3$  dimensional matrices ( $n$  is the cardinality of the state space  $\mathcal{X}$ ), whose  $i$ th columns are  $\phi_v^{(i)} = (\phi_v^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_2$  and  $\phi_u^{(i)} = (\phi_u^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_3$ . We make the following standard assumption as in [11]:

**(A4)** The basis functions  $\{\phi_v^{(i)}\}_{i=1}^{\kappa_2}$  and  $\{\phi_u^{(i)}\}_{i=1}^{\kappa_3}$  are linearly independent. In particular,  $\kappa_2, \kappa_3 \ll n$  and  $\Phi_v$  and  $\Phi_u$  are full rank. Moreover, for every  $v \in \mathbb{R}^{\kappa_2}$  and  $u \in \mathbb{R}^{\kappa_3}$ ,  $\Phi_v v \neq e$  and  $\Phi_u u \neq e$ , where  $e$  is the  $n$ -dimensional vector with all entries equal to one.

Let  $T_v^\theta$  and  $T_u^\theta$  denote the Bellman operators for value and square value functions of the policy governed by parameter  $\theta$ , respectively. These operators are defined as: For any  $y \in \mathbb{R}^{2n}$  such that  $y = [y_v; y_u]$  and  $y_v, y_u \in \mathbb{R}^n$ , we have

$$T_v^\theta y = \mathbf{r}^\theta + \gamma \mathbf{P}^\theta y_v, \quad (11)$$

$$T_u^\theta y = \mathbf{R}^\theta \mathbf{r}^\theta + 2\gamma \mathbf{R}^\theta \mathbf{P}^\theta y_v + \gamma^2 \mathbf{P}^\theta y_u, \quad (12)$$

where  $\mathbf{r}^\theta$  and  $\mathbf{P}^\theta$  are the reward vector and the transition probability matrix of policy  $\theta$ , and  $\mathbf{R}^\theta = \text{diag}(\mathbf{r}^\theta)$ .

Let  $[\Phi_v \bar{v}; \Phi_u \bar{u}]$  denote the unique fixed-point of the projected Bellman operator  $\Pi T$ , i.e.,

$$\Phi_v \bar{v} = \Pi_v(T_v \bar{v}), \text{ and } \Phi_u \bar{u} = \Pi_u(T_u \bar{u}), \quad (13)$$

where  $\Pi_v$  and  $\Pi_u$  project into the linear spaces spanned by the columns of  $\Phi_v$  and  $\Phi_u$ , respectively.

We now describe the TD algorithm that updates the critic parameters  $(v_t, u_t)$  corresponding to the value and square value functions (Note that we require critic estimates for both the unperturbed as well as the perturbed policy parameters)<sup>3</sup>:

**Critic Update:** Calculate the temporal difference (TD)-errors  $\delta_t, \delta_t^+$  for the value and  $\epsilon_t, \epsilon_t^+$  for the square value functions using (15), and update the critic parameters  $v_t, v_t^+$  for the value and  $u_t, u_t^+$  for the square value functions as follows:

$$\begin{aligned} \text{Unperturbed: } v_{t+1} &= v_t + \zeta_3(t) \delta_t \phi_v(x_t), & u_{t+1} &= u_t + \zeta_3(t) \epsilon_t \phi_u(x_t), \\ \text{Perturbed: } v_{t+1}^+ &= v_t^+ + \zeta_3(t) \delta_t^+ \phi_v(x_t^+), & u_{t+1}^+ &= u_t^+ + \zeta_3(t) \epsilon_t^+ \phi_u(x_t^+), \end{aligned} \quad (14)$$

where the TD-errors  $\delta_t, \delta_t^+, \epsilon_t, \epsilon_t^+$  in (14) are computed as

$$\begin{aligned} \text{Unperturbed: } \delta_t &= R(x_t, a_t) + \gamma v_t^\top \phi_v(x_{t+1}) - v_t^\top \phi_v(x_t), \\ \epsilon_t &= R(x_t, a_t)^2 + 2\gamma R(x_t, a_t) v_t^\top \phi_v(x_{t+1}) + \gamma^2 u_t^\top \phi_u(x_{t+1}) - u_t^\top \phi_u(x_t), \\ \text{Perturbed: } \delta_t^+ &= R(x_t^+, a_t^+) + \gamma v_t^{+\top} \phi_v(x_{t+1}^+) - v_t^{+\top} \phi_v(x_t^+), \\ \epsilon_t^+ &= R(x_t^+, a_t^+)^2 + 2\gamma R(x_t^+, a_t^+) v_t^{+\top} \phi_v(x_{t+1}^+) + \gamma^2 u_t^{+\top} \phi_u(x_{t+1}^+) \\ &\quad - u_t^{+\top} \phi_u(x_t^+). \end{aligned} \quad (15)$$

<sup>3</sup>This algorithm is a straightforward extension of the algorithm proposed by [48] to the discounted setting.

Note that the TD-error  $\epsilon$  for the square value function  $U$  comes directly from its Bellman equation (2). Theorem 2 (see Section 4.5) establishes that the critic parameters  $(v_t, u_t)$  governed by (14) converge to the solutions  $(\bar{v}, \bar{u})$  of the fixed point equation (13).

### 4.3 First-Order Algorithms: RS-SPSA-G and RS-SF-G

**SPSA**-based estimate for  $\nabla V^\theta(x^0)$ , and similarly for  $\nabla U^\theta(x^0)$ , is given by

$$\nabla_i \widehat{V}^\theta(x^0) \approx \frac{\widehat{V}^{\theta+\beta\Delta}(x^0) - \widehat{V}^\theta(x^0)}{\beta\Delta^{(i)}}, \quad i = 1, \dots, \kappa_1, \quad (16)$$

where  $\Delta$  is a vector of independent Rademacher random variables. The advantage of this estimator is that it perturbs all directions at the same time (the numerator is identical in all  $\kappa_1$  components). So, the number of function measurements needed for this estimator is always two, independent of the dimension  $\kappa_1$ . However, unlike the SPSA estimates in [38] that use two-sided balanced estimates (simulations with parameters  $\theta - \beta\Delta$  and  $\theta + \beta\Delta$ ), our gradient estimates are one-sided (simulations with parameters  $\theta$  and  $\theta + \beta\Delta$ ) and resemble those in [19]. The use of one-sided estimates is primarily because the updates of the Lagrangian parameter  $\lambda$  require a simulation with the running parameter  $\theta$ . Using a balanced gradient estimate would therefore come at the cost of an additional simulation (the resulting procedure would then require three simulations), which we avoid by using one-sided gradient estimates.

**SF**-based method estimates not the gradient of a function  $H(\theta)$  itself, but rather the convolution of  $\nabla H(\theta)$  with the Gaussian density function  $\mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I})$ , i.e.,

$$\begin{aligned} C_\beta H(\theta) &= \int \mathcal{G}_\beta(\theta - z) \nabla_z H(z) dz = \int \nabla_z \mathcal{G}_\beta(z) H(\theta - z) dz \\ &= \frac{1}{\beta} \int -z' \mathcal{G}_1(z') H(\theta - \beta z') dz', \end{aligned}$$

where  $\mathcal{G}_\beta$  is a  $\kappa_1$ -dimensional p.d.f. The first equality above follows by using integration by parts and the second one by using the fact that  $\nabla_z \mathcal{G}_\beta(z) = \frac{-z}{\beta^2} \mathcal{G}_\beta(z)$  and by substituting  $z' = z/\beta$ . As  $\beta \rightarrow 0$ , it can be seen that  $C_\beta H(\theta)$  converges to  $\nabla_\theta H(\theta)$  (see Chapter 6 of [13]). Thus, a one-sided SF estimate of  $\nabla V^\theta(x^0)$  is given by

$$\nabla_i \widehat{V}^\theta(x^0) \approx \frac{\Delta^{(i)}}{\beta} \left( \widehat{V}^{\theta+\beta\Delta}(x^0) - \widehat{V}^\theta(x^0) \right), \quad i = 1, \dots, \kappa_1, \quad (17)$$

where  $\Delta$  is a vector of independent Gaussian  $\mathcal{N}(0, 1)$  random variables.

**Actor Update:** Estimate the gradients  $\nabla V^\theta(x^0)$  and  $\nabla U^\theta(x^0)$  using SPSA (16) or SF (17) and update the policy parameter  $\theta$  as follows: For  $i = 1, \dots, \kappa_1$ ,

#### RS-SPSA-G:

$$\begin{aligned} \theta_{t+1}^{(i)} &= \Gamma_i \left[ \theta_t^{(i)} + \frac{\zeta_2(t)}{\beta \Delta_t^{(i)}} \left( (1 + 2\lambda_t v_t^\top \phi_v(x^0)) (v_t^+ - v_t)^\top \phi_v(x^0) \right. \right. \\ &\quad \left. \left. - \lambda_t (u_t^+ - u_t)^\top \phi_u(x^0) \right) \right], \end{aligned} \quad (18)$$

#### RS-SF-G:

$$\begin{aligned} \theta_{t+1}^{(i)} &= \Gamma_i \left[ \theta_t^{(i)} + \frac{\zeta_2(t) \Delta_t^{(i)}}{\beta} \left( (1 + 2\lambda_t v_t^\top \phi_v(x^0)) (v_t^+ - v_t)^\top \phi_v(x^0) \right. \right. \\ &\quad \left. \left. - \lambda_t (u_t^+ - u_t)^\top \phi_u(x^0) \right) \right]. \end{aligned} \quad (19)$$

For both SPSA and SF variants, the Lagrange multiplier  $\lambda$  is updated as follows:

$$\lambda_{t+1} = \Gamma_\lambda \left[ \lambda_t + \zeta_1(t) \left( u_t^\top \phi_u(x^0) - (v_t^\top \phi_v(x^0))^2 - \alpha \right) \right]. \quad (20)$$

In the above, note the following:

- 1)  $\beta > 0$  is a small fixed constant and  $\Delta_t^{(i)}$ 's are independent Rademacher and Gaussian  $\mathcal{N}(0, 1)$  random variables in SPSA and SF updates, respectively,
- 2)  $\Gamma$  is an operator that projects a vector  $\theta \in \mathbb{R}^{\kappa_1}$  to the closest point in a compact and convex set  $\Theta \subset \mathbb{R}^{\kappa_1}$ , and  $\Gamma_\lambda$  is a projection operator to  $[0, \lambda_{\max}]$ . These projection operators are necessary to keep the iterates stable and hence, ensure convergence of the algorithms.

In Section 4.5, we provide a proof of convergence of the first-order SPSA and SF algorithms to a (local) saddle point of the risk-sensitive objective function  $\widehat{L}(\theta, \lambda) \triangleq -\widehat{V}^\theta(x^0) + \lambda(\widehat{\Lambda}^\theta(x^0) - \alpha)$ .

#### 4.4 Second-Order Algorithms: RS-SPSA-N and RS-SF-N

Recall from Section 4.1 that a second-order scheme updates the policy parameter in the following manner:

$$\theta_{t+1} = \Gamma[\theta_t - \zeta_2(t) \nabla_\theta^2 L(\theta, \lambda)^{-1} \nabla_\theta L(\theta, \lambda)]. \quad (21)$$

From the above, it is evident that for any second-order method, an estimate of the Hessian  $\nabla_\theta^2 L(\theta, \lambda)$  of the Lagrangian is necessary, in addition to an estimate of the gradient  $\nabla_\theta L(\theta, \lambda)$ . As in the case of the gradient based schemes outlined earlier, we employ the simultaneous perturbation technique to develop these estimates. The first algorithm, henceforth referred to as RS-SPSA-N, uses SPSA for the gradient/Hessian estimates. On the other hand, the second algorithm, henceforth referred to as RS-SF-N, uses a smoothed functional (SF) approach for the gradient/Hessian estimates.

##### 4.4.1 RS-SPSA-N Algorithm

The Hessian w.r.t.  $\theta$  of  $L(\theta, \lambda)$  can be written as follows:

$$\begin{aligned} \nabla_\theta^2 L(\theta, \lambda) &= -\nabla_\theta^2 V^\theta(x^0) + \lambda \nabla_\theta^2 \Lambda^\theta(x^0) \\ &= -\nabla^2 V^\theta(x^0) + \lambda (\nabla^2 U^\theta(x^0) - 2V^\theta(x^0) \nabla^2 V^\theta(x^0) - 2\nabla V^\theta(x^0) \nabla V^\theta(x^0)^\top). \end{aligned} \quad (22)$$

**Critic Update:** As in the case of the gradient based schemes, we run two simulations. However, perturbed simulation here corresponds to the policy parameter  $\theta + \beta(\Delta + \widehat{\Delta})$ , where  $\Delta$  and  $\widehat{\Delta}$  represent vectors of independent  $\kappa_1$ -dimensional Rademacher random variables. The critic parameters  $v_t, u_t$  from unperturbed simulation and  $v_t^+, u_t^+$  from perturbed simulation are updated as described earlier in Section 4.2.

**Gradient and Hessian Estimates:** Using an SPSA-based estimation technique (see Chapter 7 of [13]), the gradient and Hessian of the value function  $V$ , and similarly of the square value function  $U$ , are estimated as follows: For  $i = 1, \dots, \kappa_1$ ,

$$\begin{aligned} \nabla_i \widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta \Delta^{(i)}} = \frac{(v_t^+ - v_t)^\top \phi_v(x^0)}{\beta \Delta^{(i)}}, \\ \nabla_{i,j}^2 \widehat{V}^\theta(x^0) &\approx \frac{\widehat{V}^{\theta+\beta(\Delta+\widehat{\Delta})}(x^0) - \widehat{V}^\theta(x^0)}{\beta^2 \Delta^{(i)} \widehat{\Delta}^{(j)}} = \frac{(v_t^+ - v_t)^\top \phi_v(x^0)}{\beta^2 \Delta^{(i)} \widehat{\Delta}^{(j)}}. \end{aligned}$$

The correctness of the above estimates in the limit as  $\beta \rightarrow 0$  can be inferred from Lemma 12 in the Appendix. The main idea is to expand using suitable Taylor expansions and observe that the bias terms vanish as  $\Delta$ , being Rademacher, are zero-mean. As in the case of RS-SPSA, this is an one-sided estimate with the unperturbed simulation required for updating the Lagrange multiplier.

**Hessian Update:** Using the critic values from the two simulations, we estimate the Hessian  $\nabla_{\theta}^2 L(\theta, \lambda)$  as follows: Let  $H_t^{(i,j)}$  denote the  $t$ th estimate of the  $(i, j)$ th element of the Hessian. Then, for  $i, j = 1, \dots, \kappa_1$ , with  $i \leq j$ , the update is

$$H_{t+1}^{(i,j)} = H_t^{(i,j)} + \zeta_2(t) \left[ \frac{(1 + \lambda_t(v_t + v_t^+)^\top \phi_v(x^0))(v_t - v_t^+)^\top \phi_v(x^0)}{\beta^2 \Delta_t^{(i)} \widehat{\Delta}_t^{(j)}} + \frac{\lambda_t(u_t^+ - u_t)^\top \phi_u(x^0)}{\beta^2 \Delta_t^{(i)} \widehat{\Delta}_t^{(j)}} - H_t^{(i,j)} \right], \quad (23)$$

and for  $i > j$ , we simply set  $H_{t+1}^{(i,j)} = H_{t+1}^{(j,i)}$ . Finally, we set  $H_{t+1} = \Upsilon([H_{t+1}^{(i,j)}]_{i,j=1}^{\kappa_1})$ , where  $\Upsilon(\cdot)$  denotes an operator that projects a square matrix onto the set of symmetric and positive definite matrices. This projection is a standard requirement to ensure convergence of  $H$  to the Hessian  $\nabla_{\theta}^2 L(\theta, \lambda)$ .

**Actor Update:** Let  $M_t \triangleq H_t^{-1}$  denote the inverse of the the Hessian estimate  $H_t$ . We incorporate a Newton decrement to update the policy parameter  $\theta$  as follows:

$$\theta_{t+1}^{(i)} = \Gamma_i \left[ \theta_t^{(i)} + \zeta_2(t) \sum_{j=1}^{\kappa_1} M_t^{(i,j)} \left( \frac{(1 + 2\lambda_t v_t^\top \phi_v(x^0))(v_t^+ - v_t)^\top \phi_v(x^0)}{\beta \Delta_t^{(j)}} - \frac{\lambda_t(u_t^+ - u_t)^\top \phi_u(x^0)}{\beta \Delta_t^{(j)}} \right) \right]. \quad (24)$$

In the long run,  $M_t$  converges to  $\nabla_{\theta}^2 L(\theta, \lambda)^{-1}$ , while the last term in the brackets in (24) converges to  $\nabla_{\theta} L(\theta, \lambda)$  and hence, the update (24) can be seen to descend in  $\theta$  using a Newton decrement. Note that the Lagrange multiplier update here is the same as that in RS-SPSA-G.

#### 4.4.2 RS-SF-N Algorithm

**Gradient and Hessian Estimates:** While the gradient estimate here is the same as that in the RS-SF-G algorithm, the Hessian is estimated as follows: Recall that  $\Delta = (\Delta^{(1)}, \dots, \Delta^{(\kappa_1)})^\top$  is a vector of mutually independent  $\mathcal{N}(0, 1)$  random variables. Let  $\bar{H}(\Delta)$  be a  $\kappa_1 \times \kappa_1$  matrix defined as

$$\bar{H}(\Delta) \triangleq \begin{bmatrix} (\Delta^{(1)^2} - 1) & \Delta^{(1)} \Delta^{(2)} & \dots & \Delta^{(1)} \Delta^{(\kappa_1)} \\ \Delta^{(2)} \Delta^{(1)} & (\Delta^{(2)^2} - 1) & \dots & \Delta^{(2)} \Delta^{(\kappa_1)} \\ \dots & \dots & \dots & \dots \\ \Delta^{(\kappa_1)} \Delta^{(1)} & \Delta^{(\kappa_1)} \Delta^{(2)} & \dots & (\Delta^{(\kappa_1)^2} - 1) \end{bmatrix}. \quad (25)$$

Then, the Hessian  $\nabla_{\theta}^2 L(\theta, \lambda)$  is approximated as

$$\nabla_{\theta}^2 L(\theta, \lambda) \approx \frac{1}{\beta^2} \left[ \bar{H}(\Delta) (L(\theta + \beta \Delta, \lambda) - L(\theta, \lambda)) \right]. \quad (26)$$

The correctness of the above estimate in the limit as  $\beta \rightarrow 0$  can be seen from Lemma 13 in the Appendix. The main idea involves convolving the Hessian with a Gaussian density function (similar to RS-SF) and then performing integration by parts twice.

**Critic Update:** As in the case of the RS-SF-G algorithm, we run two simulations with unperturbed and perturbed policy parameters, respectively. Recall that the perturbed simulation corresponds to the policy parameter  $\theta + \beta \Delta$ ,

where  $\Delta$  represent a vector of independent  $\kappa_1$ -dimensional Gaussian  $\mathcal{N}(0, 1)$  random variables. The critic parameters for both these simulations are updated as described earlier in Section 4.2.

**Hessian Update:** As in RS-SPSA-N, let  $H_t^{(i,j)}$  denote the  $(i, j)$ th element of the Hessian estimate  $H_t$  at time step  $t$ . Using (26), we devise the following update rule for the Hessian estimate  $H_t$ : For  $i, j, k = 1, \dots, \kappa_1, j < k$ , the update is

$$H_{t+1}^{(i,i)} = H_t^{(i,i)} + \zeta_2(t) \left[ \frac{(\Delta_t^{(i)^2} - 1)}{\beta^2} \left( (1 + \lambda_t(v_t + v_t^+)^\top \phi_v(x^0))(v_t - v_t^+)^\top \phi_v(x^0) + \lambda_t(u_t^+ - u_t)^\top \phi_u(x^0) \right) - H_t^{(i,i)} \right], \quad (27)$$

$$H_{t+1}^{(j,k)} = H_t^{(j,k)} + \zeta_2(t) \left[ \frac{\Delta_t^{(j)} \Delta_t^{(k)}}{\beta^2} \left( (1 + \lambda_t(v_t + v_t^+)^\top \phi_v(x^0))(v_t - v_t^+)^\top \phi_v(x^0) + \lambda_t(u_t^+ - u_t)^\top \phi_u(x^0) \right) - H_t^{(j,k)} \right], \quad (28)$$

and for  $j > k$ , we set  $H_{t+1}^{(j,k)} = H_{t+1}^{(k,j)}$ . As before, we set  $H_{t+1} = \Upsilon([H_{t+1}^{(i,j)}]_{i,j=1}^{\kappa_1})$  and let  $M_{t+1} \triangleq H_{t+1}^{-1}$  denote its inverse.

**Actor Update:** Using the gradient and Hessian estimates from the above, we update the policy parameter  $\theta$  as follows:

$$\theta_{t+1}^{(i)} = \Gamma_i \left[ \theta_t^{(i)} + \zeta_2(t) \sum_{j=1}^{\kappa_1} M_t^{(i,j)} \frac{\Delta_t^{(j)}}{\beta} \left( (1 + 2\lambda_t v_t^\top \phi_v(x^0))(v_t^+ - v_t)^\top \phi_v(x^0) - \lambda_t(u_t^+ - u_t)^\top \phi_u(x^0) \right) \right]. \quad (29)$$

As in the case of RS-SPSA-N, it can be seen that the above update rule is equivalent to descent with a Newton decrement, since  $M_t$  converges to  $\nabla_\theta^2 L(\theta, \lambda)^{-1}$ , and the last term in the brackets in (29) converges to  $\nabla_\theta L(\theta, \lambda)$ . The Lagrange multiplier  $\lambda$  update here is the same as that in RS-SF-G.

## 4.5 Convergence Analysis of the Risk-Sensitive Actor-Critic Algorithms

Our proposed actor-critic algorithms use multi-timescale stochastic approximation and we use the ordinary differential equation (ODE) approach (see Chapter 6 of [17]) to analyze their convergence. We first provide the analysis for the first-order algorithms in Section 4.5.1 and later analyze the second-order algorithms in Section 4.5.2.

### 4.5.1 Convergence of the First-Order Algorithms: RS-SPSA-G and RS-SF-G

The proof of convergence of the RS-SPSA-G and RS-SF-G algorithms to a (local) saddle point of the risk-sensitive objective function  $\hat{L}(\theta, \lambda) \triangleq -\hat{V}^\theta(x^0) + \lambda(\hat{\Lambda}^\theta(x^0) - \alpha) = -\hat{V}^\theta(x^0) + \lambda(\hat{U}^\theta(x^0) - \hat{V}^\theta(x^0)^2 - \alpha)$  contains the following three main steps - critic convergence, analysis of  $\theta$  and  $\lambda$  recursions. Note that since RS-SPSA-G and RS-SF-G use different methods to estimate the gradient, their proofs only differ in the second step, i.e., the convergence of the policy parameter  $\theta$ .

**Step 1: (Critic's Convergence)** Since the critic's update is on the fastest time-scale and the step-size schedules satisfy (A4), we can assume in this analysis that  $\theta$  and  $\lambda$  are time-invariant quantities. The following theorem shows that the value and square value estimates of policies  $\theta$  and  $\theta^+ = \theta + \beta\Delta$  converge.

**Theorem 2.** Under (A1)-(A4), for any given policy parameter  $\theta$  and Lagrange multiplier  $\lambda$ , the critic parameters  $\{v_t\}, \{v_t^+\}$  and  $\{u_t\}, \{u_t^+\}$  governed by the recursions of (14) converge, i.e.,  $v_t \rightarrow \bar{v}, v_t^+ \rightarrow \bar{v}^+$  and  $u_t \rightarrow \bar{u}, u_t^+ \rightarrow \bar{u}^+$ , where  $\bar{v}, \bar{v}^+$  and  $\bar{u}, \bar{u}^+$  are the unique solutions to

$$\begin{aligned} (\Phi_v^\top \mathbf{D}_\gamma^\theta \Phi_v) \bar{v} &= \Phi_v^\top \mathbf{D}_\gamma^\theta T_v^\theta [\Phi_v \bar{v}], & (\Phi_v^\top \mathbf{D}_\gamma^{\theta^+} \Phi_v) \bar{v}^+ &= \Phi_v^\top \mathbf{D}_\gamma^{\theta^+} T_v^{\theta^+} [\Phi_v \bar{v}^+], \\ (\Phi_u^\top \mathbf{D}_\gamma^\theta \Phi_u) \bar{u} &= \Phi_u^\top \mathbf{D}_\gamma^\theta T_u^\theta [\Phi_u \bar{u}], & (\Phi_u^\top \mathbf{D}_\gamma^{\theta^+} \Phi_u) \bar{u}^+ &= \Phi_u^\top \mathbf{D}_\gamma^{\theta^+} T_u^{\theta^+} [\Phi_u \bar{u}^+], \end{aligned}$$

where  $n$  is the total number states in the state space  $\mathcal{X}$ , and

- $\Phi_v$  and  $\Phi_u$  are  $n \times \kappa_2$  and  $n \times \kappa_3$  dimensional matrices ( $\kappa_2, \kappa_3 \ll n$ ) whose  $i$ 'th columns are  $\phi_v^{(i)} = (\phi_v^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_2$  and  $\phi_u^{(i)} = (\phi_u^{(i)}(x), x \in \mathcal{X})^\top$ ,  $i = 1, \dots, \kappa_3$ .
- $\mathbf{D}_\gamma^\theta$  and  $\mathbf{D}_\gamma^{\theta^+}$  denote the diagonal matrices with entries  $d_\gamma^\theta(x)$  and  $d_\gamma^{\theta^+}(x)$  for all  $x \in \mathcal{X}$ .
- $T_v^\theta, T_v^{\theta^+}$  and  $T_u^\theta, T_u^{\theta^+}$  are the Bellman operators for value and square value functions of policies  $\theta$  and  $\theta^+$ , respectively. For any  $y \in \mathbb{R}^{2n}$  such that  $y = [y_v; y_u]$  and  $y_v, y_u \in \mathbb{R}^n$ , these operators are defined as  $T_v^\theta y = \mathbf{r}^\theta + \gamma \mathbf{P}^\theta y_v$  and  $T_u^\theta y = \mathbf{R}^\theta \mathbf{r}^\theta + 2\gamma \mathbf{R}^\theta \mathbf{P}^\theta y_v + \gamma^2 \mathbf{P}^\theta y_u$ , where  $\mathbf{r}^\theta$  and  $\mathbf{P}^\theta$  are the reward vector and the transition probability matrix of policy  $\theta$ , and  $\mathbf{R}^\theta = \text{diag}(\mathbf{r}^\theta)$ .

*Proof.* See Appendix A. □

**Step 2: (Analysis of  $\theta$ -recursion)** Due to timescale separation, the value of  $\lambda$  (updated on a slower timescale) is assumed to be constant for the analysis of the  $\theta$ -update. In the following, we show that the update of  $\theta$  is equivalent to gradient descent for the function  $\hat{L}(\theta, \lambda)$  and converges to a limiting set that depends on  $\lambda$ .

Consider the following ODE

$$\dot{\theta}_t = \check{\Gamma} \left( \nabla_\theta \hat{L}(\theta_t, \lambda) \right), \quad (30)$$

where  $\check{\Gamma}$  is defined as follows: For any bounded continuous function  $f(\cdot)$ ,

$$\check{\Gamma}(f(\theta_t)) = \lim_{\tau \rightarrow 0} \frac{\Gamma(\theta_t + \tau f(\theta_t)) - \theta_t}{\tau}. \quad (31)$$

The projection operator  $\check{\Gamma}(\cdot)$  ensures that the evolution of  $\theta$  via the ODE (30) stays within the bounded set  $C \in \mathbb{R}^{\kappa_1}$ .

Let  $\mathcal{Z}_\lambda = \{\theta \in C : \check{\Gamma}(\nabla_\theta \hat{L}(\theta, \lambda)) = 0\}$  denote the set of asymptotically stable equilibrium points of the ODE (30) and  $\mathcal{Z}_\lambda^\varepsilon = \{\theta \in C : \|\theta - \theta_0\| < \varepsilon, \theta_0 \in \mathcal{Z}_\lambda\}$  denote the set of points in the  $\varepsilon$ -neighborhood of  $\mathcal{Z}_\lambda$ . The main result regarding the convergence of the policy parameter  $\theta$  for both the RS-SPSA-G and RS-SF-G algorithms is as follows:

**Theorem 3.** Under (A1)-(A4), for any given Lagrange multiplier  $\lambda$  and  $\varepsilon > 0$ , there exists  $\beta_0 > 0$  such that for all  $\beta \in (0, \beta_0)$ ,  $\theta_t \rightarrow \theta^* \in \mathcal{Z}_\lambda^\varepsilon$  almost surely.

*Proof.* See Appendix A. ■ □

**Step 3: (Analysis of  $\lambda$ -recursion and Convergence to a Local Saddle Point)** We first show that the  $\lambda$ -recursion converges and then prove that the whole algorithm converges to a local saddle point of  $\hat{L}(\theta, \lambda)$ .

We define the following ODE governing the evolution of  $\lambda$ :

$$\dot{\lambda}_t = \check{\Gamma}_\lambda [\hat{L}^{\theta_t}(x^0) - \alpha] = \check{\Gamma}_\lambda [\hat{U}^{\theta_t}(x^0) - \hat{V}^{\theta_t}(x^0)^2 - \alpha], \quad (32)$$

where  $\check{\Gamma}_\lambda$  is defined as follows: For any bounded continuous function  $f(\cdot)$ ,

$$\check{\Gamma}_\lambda(f(\lambda_t)) = \lim_{\tau \rightarrow 0} \frac{\Gamma(\lambda_t + \tau f(\lambda_t)) - \lambda_t}{\tau}. \quad (33)$$

The operator  $\check{\Gamma}_\lambda$  is similar to the operator  $\check{\Gamma}$  defined in (31).

**Theorem 4.**  $\lambda_t \rightarrow \mathcal{F}$  almost surely as  $t \rightarrow \infty$ , where  $\mathcal{F} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}], \check{\Gamma}_\lambda[\hat{\Lambda}^{\theta^\lambda}(x^0) - \alpha] = 0, \theta^\lambda \in \mathcal{Z}_\lambda\}$ .

*Proof.* The proof follows the same steps as in Theorem 3 in [8]. ■ □

The last step is to establish that the algorithm converges to a (local) saddle point of  $\hat{L}(\theta, \lambda)$ . In other words, it converges to a pair  $(\theta^*, \lambda^*)$  that are a local minimum w.r.t.  $\theta$  and a local maximum w.r.t.  $\lambda$  of  $\hat{L}(\theta, \lambda)$ . From Theorem 4,  $\lambda_t \rightarrow \lambda^*$  for some  $\lambda^* \in [0, \lambda_{\max}]$  such that  $\theta^{\lambda^*} \in \mathcal{Z}_{\lambda^*}$  and  $\check{\Gamma}_{\lambda^*}[\hat{\Lambda}^{\theta^{\lambda^*}}(x^0) - \alpha] = 0$ . As in [16], we invoke the envelope theorem of mathematical economics [29] to conclude that the ODE  $\dot{\lambda}_t = \check{\Gamma}_\lambda[\hat{\Lambda}^{\theta_t}(x^0) - \alpha]$  is equivalent to  $\dot{\lambda}_t = \check{\Gamma}_\lambda[\nabla_\lambda \hat{L}(\theta^{\lambda^*}, \lambda^*)]$ . From the above, it is clear that  $(\theta_t, \lambda_t)$  governed by (18)–(20) converges to a local saddle point of  $\hat{L}(\theta, \lambda)$ .

#### 4.5.2 Convergence of the Second-Order Algorithms: RS-SPSA-N and RS-SF-N

Convergence analysis of the second-order algorithms involves the same steps as that of the first-order algorithms. In particular, the first step involving the TD-critic and the third step involving the analysis of  $\lambda$ -recursion follow along similar lines as earlier, whereas  $\theta$ -recursion analysis in the second step differs significantly.

**Step 2: (Analysis of  $\theta$ -recursion for RS-SPSA-N and RS-SF-N)** Since the policy parameter is updated in the descent direction with a Newton decrement, the limiting ODE of the  $\theta$ -recursion for the second order algorithms is given by

$$\dot{\theta}_t = \check{\Gamma} \left( \Upsilon \left( \nabla_\theta^2 L(\theta_t, \lambda) \right)^{-1} \nabla_\theta L(\theta_t, \lambda) \right), \quad (34)$$

where  $\check{\Gamma}$  is as before (see (31)). Let

$$\mathcal{Z}_\lambda = \left\{ \theta \in C : -\nabla_\theta L(\theta_t, \lambda)^T \Upsilon \left( \nabla_\theta^2 L(\theta_t, \lambda) \right)^{-1} \nabla_\theta L(\theta_t, \lambda) = 0 \right\}.$$

denote the set of asymptotically stable equilibrium points of the ODE (34) and  $\mathcal{Z}_\lambda^\varepsilon$  its  $\varepsilon$ -neighborhood. Then, we have the following analogue of Theorem 3 for the RS-SPSA-N and RS-SF-N algorithms:

**Theorem 5.** Under (A1)–(A4), for any given Lagrange multiplier  $\lambda$  and  $\varepsilon > 0$ , there exists  $\beta_0 > 0$  such that for all  $\beta \in (0, \beta_0)$ ,  $\theta_t \rightarrow \theta^* \in \mathcal{Z}_\lambda^\varepsilon$  almost surely.

*Proof.* See Appendix A. ■ □

**Remark 1.** In the above, we established asymptotic limits for all our algorithms using the ODE approach. To the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms. This is true even for the actor-critic algorithms that do not incorporate any risk criterion. It would be an interesting direction for future research to obtain finite-time bounds on the quality of the solution obtained by these algorithms.

## 4.6 Extension of the Algorithms to Sharpe Ratio Optimization

The gradient of Sharpe ratio (SR),  $S(\theta)$ , in the discounted setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{\Lambda^\theta(x^0)}} \left( \nabla V^\theta(x^0) - \frac{V^\theta(x^0)}{2\Lambda^\theta(x^0)} \nabla \Lambda^\theta(x^0) \right).$$

The actor recursions for the variants of the RS-SPSA-G and RS-SF-G algorithms that optimize the SR objective are as follows:

### RS-SPSA-G

$$\theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} + \frac{\zeta_2(t)}{\sqrt{u_t^\top \phi_u(x^0) - (v_t^\top \phi_v(x^0))^2 \beta \Delta_t^{(i)}}} \left( (v_t^+ - v_t)^\top \phi_v(x^0) - \frac{v_t^\top \phi_v(x^0) ((u_t^+ - u_t)^\top \phi_u(x^0) - 2v_t^\top \phi_v(x^0)(v_t^+ - v_t)^\top \phi_v(x^0))}{2(u_t^\top \phi_u(x^0) - (v_t^\top \phi_v(x^0))^2)} \right) \right). \quad (35)$$

### RS-SF-G

$$\theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} + \frac{\zeta_2(t) \Delta_t^{(i)}}{\beta \sqrt{u_t^\top \phi_u(x^0) - (v_t^\top \phi_v(x^0))^2}} \left( (v_t^+ - v_t)^\top \phi_v(x^0) - \frac{v_t^\top \phi_v(x^0) ((u_t^+ - u_t)^\top \phi_u(x^0) - 2v_t^\top \phi_v(x^0)(v_t^+ - v_t)^\top \phi_v(x^0))}{2(u_t^\top \phi_u(x^0) - (v_t^\top \phi_v(x^0))^2)} \right) \right). \quad (36)$$

Note that only the actor recursion changes for SR optimization, while the rest of the updates that include the critic recursions for nominal and perturbed parameters remain the same as before in the SPSA and SF based algorithms. Further, SR optimization does not involve the Lagrange parameter  $\lambda$ , and thus, the proposed actor-critic algorithms are two time-scale (instead of three time-scale as in the described algorithms) stochastic approximation algorithms in this case.

**Remark 2.** For the SR objective, the proposed algorithms can be modified to work with only one simulated trajectory of the system. This is because in the SR case, we do not require the Lagrange multiplier  $\lambda$ , and thus, the simulated trajectory corresponding to the nominal policy parameter  $\theta$  is not necessary. In this implementation, the gradient is estimated as  $\nabla_i S(\theta) \approx S(\theta + \beta \Delta) / \beta \Delta^{(i)}$  for SPSA and as  $\nabla_i S(\theta) \approx (\Delta^{(i)} / \beta) S(\theta + \beta \Delta)$  for SF.

**Remark 3.** The second-order variants of the algorithms for SR optimization can be worked out along similar lines as outlined in Section 4.4 and the details are omitted here.

## 5 Average Reward Setting

The average reward under policy  $\mu$  is defined as

$$\rho(\mu) = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=0}^{T-1} R_t \mid \mu \right] = \sum_{x,a} d^\mu(x) \mu(a|x) r(x,a) = \sum_{x,a} \pi^\mu(x,a) r(x,a),$$

where  $d^\mu$  and  $\pi^\mu$  are the stationary distributions of policy  $\mu$  over states and state-action pairs, respectively (see Section 2). The goal in the standard (risk-neutral) average reward formulation is to find an *average optimal* policy, i.e.,  $\mu^* = \arg \max_\mu \rho(\mu)$ . For all states  $x \in \mathcal{X}$  and actions  $a \in \mathcal{A}$ , the *differential* action-value and value functions of policy  $\mu$  are defined respectively as

$$Q^\mu(x,a) = \sum_{t=0}^{\infty} \mathbb{E}[R_t - \rho(\mu) \mid x_0 = x, a_0 = a, \mu],$$

$$V^\mu(x) = \sum_a \mu(a|x) Q^\mu(x,a).$$

These functions satisfy the following Poisson equations [35]

$$\rho(\mu) + V^\mu(x) = \sum_a \mu(a|x) [r(x,a) + \sum_{x'} P(x'|x,a) V^\mu(x')], \quad (37)$$

$$\rho(\mu) + Q^\mu(x,a) = r(x,a) + \sum_{x'} P(x'|x,a) V^\mu(x'). \quad (38)$$



In the context of risk-sensitive MDPs, different criteria have been proposed to define a measure of *variability* in the average reward setting, among which we consider the *long-run variance* of  $\mu$  [21] defined as

$$\Lambda(\mu) = \sum_{x,a} \pi^\mu(x,a) [r(x,a) - \rho(\mu)]^2 = \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=0}^{T-1} (R_t - \rho(\mu))^2 \mid \mu \right]. \quad (39)$$

This notion of variability is based on the observation that it is the frequency of occurrence of state-action pairs that determine the variability in the average reward. It is easy to show that

$$\Lambda(\mu) = \eta(\mu) - \rho(\mu)^2, \quad \text{where} \quad \eta(\mu) = \sum_{x,a} \pi^\mu(x,a) r(x,a)^2.$$

We consider the following risk-sensitive measure for average reward MDPs in this paper:

$$\max_{\theta} \rho(\theta) \quad \text{subject to} \quad \Lambda(\theta) \leq \alpha, \quad (40)$$

for a given  $\alpha > 0$ .<sup>4</sup> As in the discounted setting, we employ the Lagrangian relaxation procedure to convert (40) to the unconstrained problem

$$\max_{\lambda} \min_{\theta} \left( L(\theta, \lambda) \triangleq -\rho(\theta) + \lambda(\Lambda(\theta) - \alpha) \right).$$

As in the discounted setting, we descend in  $\theta$  using  $\nabla_{\theta} L(\theta, \lambda) = -\nabla_{\theta} \rho(\theta) + \lambda \nabla_{\theta} \Lambda(\theta)$  and ascend in  $\lambda$  using  $\nabla_{\lambda} L(\theta, \lambda) = \Lambda(\theta) - \alpha$ , to find the saddle point of  $L(\theta, \lambda)$ . Since  $\nabla_{\theta} \Lambda(\theta) = \nabla_{\theta} \eta(\theta) - 2\rho(\theta) \nabla_{\theta} \rho(\theta)$ , in order to compute  $\nabla_{\theta} \Lambda(\theta)$  it would be enough to calculate  $\nabla_{\theta} \rho(\theta)$  and  $\nabla_{\theta} \eta(\theta)$ . Let  $U^\mu$  and  $W^\mu$  denote the differential value and action-value functions associated with the square reward under policy  $\mu$ , respectively. These two quantities satisfy the following Poisson equations:

$$\begin{aligned} \eta(\mu) + U^\mu(x) &= \sum_a \mu(a|x) [r(x,a)^2 + \sum_{x'} P(x'|x,a) U^\mu(x')], \\ \eta(\mu) + W^\mu(x,a) &= r(x,a)^2 + \sum_{x'} P(x'|x,a) U^\mu(x'). \end{aligned} \quad (41)$$

The gradients of  $\rho(\theta)$  and  $\eta(\theta)$  are given by the following lemma:

**Lemma 6.** *Under (A1) and (A2), we have*

$$\nabla_{\theta} \rho(\theta) = \sum_{x,a} \pi^\theta(x,a) \nabla_{\theta} \log \mu(a|x; \theta) Q(x,a; \theta), \quad (42)$$

$$\nabla_{\theta} \eta(\theta) = \sum_{x,a} \pi^\theta(x,a) \nabla_{\theta} \log \mu(a|x; \theta) W(x,a; \theta). \quad (43)$$

*Proof.* The proof of  $\nabla_{\theta} \rho(\theta)$  can be found in the literature (e.g., [45, 25]). To prove  $\nabla_{\theta} \eta(\theta)$ , we start by the fact that from (41), we have  $U(x) = \sum_a \mu(x|a) W(x,a)$ . If we take the derivative w.r.t.  $\theta$  from both sides of this equation, we obtain

$$\begin{aligned} \nabla U(x) &= \sum_a \nabla \mu(x|a) W(x,a) + \sum_a \mu(x|a) \nabla W(x,a) \\ &= \sum_a \nabla \mu(x|a) W(x,a) + \sum_a \mu(x|a) \nabla (r(x,a)^2 - \eta + \sum_{x'} P(x'|x,a) U(x')) \\ &= \sum_a \nabla \mu(x|a) W(x,a) - \nabla \eta + \sum_{a,x'} \mu(a|x) P(x'|x,a) \nabla U(x'). \end{aligned} \quad (44)$$

<sup>4</sup>Similar to the discounted setting, the risk-sensitive average reward algorithm proposed in this paper can be easily extended to other risk measures based on the long-term variance of  $\mu$ , including the Sharpe Ratio (SR), i.e.,  $\max_{\theta} \rho(\theta) / \sqrt{\Lambda(\theta)}$ . The extension to SR will be described in more details in Section 6.2.

The second equality is by replacing  $W(x, a)$  from (41). Now if we take the weighted sum, weighted by  $d(x)$ , from both sides of (44), we have

$$\sum_x d(x) \nabla U(x) = \sum_{x,a} d(x) \nabla \mu(a|x) W(x, a) - \nabla \eta + \sum_{a,x'} d(x) \mu(a|x) P(x'|x, a) \nabla U(x'). \quad (45)$$

The claim follows from the fact that the last sum on the RHS of (45) is equal to  $\sum_x d(x) \nabla U(x)$ .  $\blacksquare$   $\square$

Note that (43) for calculating  $\nabla \eta(\theta)$  has close resemblance to (42) for  $\nabla \rho(\theta)$ , and thus, similar to what we have for (42), any function  $b : \mathcal{X} \rightarrow \mathbb{R}$  can be added or subtracted to  $W(x, a; \theta)$  on the RHS of (43) without changing the result of the integral (see e.g., [11]). So, we can replace  $W(x, a; \theta)$  with the square reward advantage function  $B(x, a; \theta) = W(x, a; \theta) - U(x; \theta)$  on the RHS of (43) in the same manner as we can replace  $Q(x, a; \theta)$  with the advantage function  $A(x, a; \theta) = Q(x, a; \theta) - V(x; \theta)$  on the RHS of (42) without changing the result of the integral. We define the temporal difference (TD) errors  $\delta_t$  and  $\epsilon_t$  for the differential value and square value functions as

$$\begin{aligned} \delta_t &= R(x_t, a_t) - \hat{\rho}_{t+1} + \hat{V}(x_{t+1}) - \hat{V}(x_t), \\ \epsilon_t &= R(x_t, a_t)^2 - \hat{\eta}_{t+1} + \hat{U}(x_{t+1}) - \hat{U}(x_t). \end{aligned}$$

If  $\hat{V}$ ,  $\hat{U}$ ,  $\hat{\rho}$ , and  $\hat{\eta}$  are unbiased estimators of  $V^\mu$ ,  $U^\mu$ ,  $\rho(\mu)$ , and  $\eta(\mu)$ , respectively, then we show in Lemma 7 that  $\delta_t$  and  $\epsilon_t$  are unbiased estimates of the advantage functions  $A^\mu$  and  $B^\mu$ , i.e.,  $\mathbb{E}[\delta_t | x_t, a_t, \mu] = A^\mu(x_t, a_t)$  and  $\mathbb{E}[\epsilon_t | x_t, a_t, \mu] = B^\mu(x_t, a_t)$ .

**Lemma 7.** *For any given policy  $\mu$ , we have*

$$\mathbb{E}[\delta_t | x_t, a_t, \mu] = A^\mu(x_t, a_t), \quad \mathbb{E}[\epsilon_t | x_t, a_t, \mu] = B^\mu(x_t, a_t).$$

*Proof.* The first statement  $\mathbb{E}[\delta_t | x_t, a_t, \mu] = A^\mu(x_t, a_t)$  has been proved in Lemma 3 of [11], so here we only prove the second statement  $\mathbb{E}[\epsilon_t | x_t, a_t, \mu] = B^\mu(x_t, a_t)$ . we may write

$$\begin{aligned} \mathbb{E}[\epsilon_t | x_t, a_t, \mu] &= \mathbb{E}[R(x_t, a_t)^2 - \hat{\eta}_{t+1} + \hat{U}(x_{t+1}) - \hat{U}(x_t) | x_t, a_t, \mu] \\ &= r(x_t, a_t)^2 - \eta(\mu) + \mathbb{E}[\hat{U}(x_{t+1}) | x_t, a_t, \mu] - U^\mu(x_t) \\ &= r(x_t, a_t)^2 - \eta(\mu) + \mathbb{E}[\mathbb{E}[\hat{U}(x_{t+1}) | x_{t+1}, \mu] | x_t, a_t] - U^\mu(x_t) \\ &= r(x_t, a_t)^2 - \eta(\mu) + \mathbb{E}[\hat{U}(x_{t+1}) | x_t, a_t] - U^\mu(x_t) \\ &= r(x_t, a_t)^2 - \eta(\mu) + \underbrace{\sum_{x_{t+1} \in \mathcal{X}} P(x_{t+1} | x_t, a_t) U^\mu(x_{t+1})}_{W^\mu(x, a)} - U^\mu(x_t) \\ &= B^\mu(x, a). \end{aligned} \quad \blacksquare$$

$\square$

From Lemma 7, we notice that  $\delta_t \psi_t$  and  $\epsilon_t \psi_t$  are unbiased estimates of  $\nabla \rho(\mu)$  and  $\nabla \eta(\mu)$ , respectively, where  $\psi_t = \psi(x_t, a_t) = \nabla \log \mu(a_t | x_t)$  is the *compatible* feature (see e.g., [45, 31]).

## 6 Average Reward Risk-Sensitive Actor-Critic Algorithm

We now present our risk-sensitive actor-critic algorithm for average reward MDPs. Algorithm 2 presents the complete structure of the algorithm along with the update rules for the average rewards  $\hat{\rho}_t, \hat{\eta}_t$ ; TD errors  $\delta_t, \epsilon_t$ ; critic  $v_t, u_t$ ; and actor  $\theta_t, \lambda_t$  parameters. The projection operators  $\Gamma$  and  $\Gamma_\lambda$  are as defined in Section 4, and similar

to the discounted setting, are necessary for the convergence proof of the algorithm. The step-size schedules satisfy (A3) defined in Section 4, plus the step size schedule  $\{\zeta_4(t)\}$  satisfies  $\zeta_4(t) = k\zeta_3(t)$ , for some positive constant  $k$ . This is to ensure that the average and critic updates are on the (same) fastest time-scale  $\{\zeta_4(t)\}$  and  $\{\zeta_3(t)\}$ , the policy parameter update is on the intermediate time-scale  $\{\zeta_2(t)\}$ , and the Lagrange multiplier update is on the slowest time-scale  $\{\zeta_1(t)\}$ . This results in a three time-scale stochastic approximation algorithm.

---

**Algorithm 2** Template of the Average Reward Risk-Sensitive Actor-Critic Algorithm

---

**Input:** parameterized policy  $\mu(\cdot|\cdot; \theta)$  and value function feature vectors  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$

**Initialization:** policy parameters  $\theta = \theta_0$ ; value function weight vectors  $v = v_0$  and  $u = u_0$ ; initial state  $x_0 \sim P_0(x)$

**for**  $t = 0, 1, 2, \dots$  **do**

    Draw action  $a_t \sim \mu(\cdot|x_t; \theta_t)$  and observe the next state  $x_{t+1} \sim P(\cdot|x_t, a_t)$  and the reward  $R(x_t, a_t)$

$$\textbf{Average Updates: } \hat{\rho}_{t+1} = (1 - \zeta_4(t))\hat{\rho}_t + \zeta_4(t)R(x_t, a_t),$$

$$\hat{\eta}_{t+1} = (1 - \zeta_4(t))\hat{\eta}_t + \zeta_4(t)R(x_t, a_t)^2$$

$$\textbf{TD Errors: } \delta_t = R(x_t, a_t) - \hat{\rho}_{t+1} + v_t^\top \phi_v(x_{t+1}) - v_t^\top \phi_v(x_t)$$

$$\epsilon_t = R(x_t, a_t)^2 - \hat{\eta}_{t+1} + u_t^\top \phi_u(x_{t+1}) - u_t^\top \phi_u(x_t)$$

$$\textbf{Critic Update: } v_{t+1} = v_t + \zeta_3(t)\delta_t\phi_v(x_t), \quad u_{t+1} = u_t + \zeta_3(t)\epsilon_t\phi_u(x_t) \quad (46)$$

$$\textbf{Actor Update: } \theta_{t+1} = \Gamma\left(\theta_t - \zeta_2(t)\left(-\delta_t\psi_t + \lambda_t(\epsilon_t\psi_t - 2\hat{\rho}_{t+1}\delta_t\psi_t)\right)\right) \quad (47)$$

$$\lambda_{t+1} = \Gamma_\lambda\left(\lambda_t + \zeta_1(t)(\hat{\eta}_{t+1} - \hat{\rho}_{t+1}^2 - \alpha)\right) \quad (48)$$

**end for**

**return** policy and value function parameters  $\theta, \lambda, v, u$

---

As in the discounted setting, the critic uses linear approximation for the differential value and square value functions, i.e.,  $\hat{V}(x) = v^\top \phi_v(x)$  and  $\hat{U}(x) = u^\top \phi_u(x)$ , where  $\phi_v(\cdot)$  and  $\phi_u(\cdot)$  are feature vectors of size  $\kappa_2$  and  $\kappa_3$ , respectively. Although our estimates of  $\rho(\theta)$  and  $\eta(\theta)$  are unbiased, since we use biased estimates for  $V^\theta$  and  $U^\theta$  (linear approximations in the critic), our gradient estimates  $\nabla_\theta \rho(\theta)$  and  $\nabla_\theta \eta(\theta)$ , and as a result  $\nabla_\theta L(\theta, \lambda)$ , are biased. The following lemma shows the bias in our estimate of  $\nabla_\theta L(\theta, \lambda)$ .

**Lemma 8.** *The bias of our actor-critic algorithm in estimating  $\nabla_\theta L(\theta, \lambda)$  for fixed  $\theta$  and  $\lambda$  is*

$$\begin{aligned} \mathcal{B}(\theta, \lambda) = \sum_x d^\theta(x) & \left( - (1 + 2\lambda\rho(\theta)) [\nabla \bar{V}^\theta(x) - \nabla v^{\theta\top} \phi_v(x)] \right. \\ & \left. + \lambda [\nabla \bar{U}^\theta(x) - \nabla u^{\theta\top} \phi_u(x)] \right), \end{aligned}$$

where  $v^{\theta\top} \phi_v(\cdot)$  and  $u^{\theta\top} \phi_u(\cdot)$  are estimates of  $V^\theta(\cdot)$  and  $U^\theta(\cdot)$  upon convergence of the TD recursion, and

$$\bar{V}^\theta(x) = \sum_a \mu(a|x) [r(x, a) - \rho(\theta) + \sum_{x'} P(x'|x, a) v^{\theta\top} \phi_v(x')],$$

$$\bar{U}^\theta(x) = \sum_a \mu(a|x) [r(x, a)^2 - \eta(\theta) + \sum_{x'} P(x'|x, a) u^{\theta\top} \phi_u(x')].$$

*Proof.* See Appendix B. ■ □

**Remark 4.** *In the discounted setting, another popular variability measure is the discounted normalized variance [21]*

$$\Lambda(\mu) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t (R_t - \rho_\gamma(\mu))^2 \right], \quad (49)$$

where  $\rho_\gamma(\mu) = \sum_{x,a} d_\gamma^\mu(x|x^0)\mu(a|x)r(x,a)$  and  $d_\gamma^\mu(x|x^0)$  is the  $\gamma$ -discounted visiting distribution of state  $x$  under policy  $\mu$ , defined in Section 2. The variability measure (49) has close resemblance to the average reward variability measure (39), and thus, any (discounted) risk measure based on (49) can be optimized similar to the corresponding average reward risk measure (39).

In the following section, we establish the convergence of our average reward actor-critic algorithm to a (local) saddle point of the risk-sensitive objective function  $L(\theta, \lambda)$ .

## 6.1 Convergence Analysis of the Average Reward Risk-Sensitive Actor-Critic Algorithm

As in the discounted setting, we use the ODE approach [17] to analyze the convergence of our average reward risk-sensitive actor-critic algorithm. The proof involves three main steps:

1. The first step is the convergence of  $\rho$ ,  $\eta$ ,  $V$ , and  $U$ , for any fixed policy  $\theta$  and Lagrange multiplier  $\lambda$ . This corresponds to a TD(0) (with extension to  $\eta$  and  $U$ ) proof. The policy and Lagrange multiplier are considered fixed because the critic's updates are on the faster time-scale than the actor's.
2. The second step is to show the convergence of  $\theta_t$  to an  $\varepsilon$ -neighborhood  $\mathcal{Z}_\lambda^\varepsilon$  of the set of asymptotically stable equilibria  $\mathcal{Z}_\lambda$  of ODE

$$\dot{\theta}_t = \check{\Gamma}(\nabla L(\theta_t, \lambda)), \quad (50)$$

where for any bounded continuous function  $f(\cdot)$ , the projection operator  $\check{\Gamma}$  is defined as

$$\check{\Gamma}(f(\theta_t)) = \lim_{\tau \rightarrow 0} \frac{\Gamma(\theta_t + \tau f(\theta_t)) - \theta_t}{\tau}. \quad (51)$$

The operator  $\check{\Gamma}$  ensures that the evolution of  $\theta$  via the ODE (50) stays within the compact and convex set  $\Theta \subset \mathbb{R}^{\kappa_1}$ . Again here it is assumed that  $\lambda$  is fixed because  $\theta$ -recursion is on a faster time-scale than  $\lambda$ 's.

3. The final step is the convergence of  $\lambda$  and showing that the whole algorithm converges to a local saddle point of  $L(\theta, \lambda)$ .

### Step 1: Critic's Convergence

**Lemma 9.** For any given policy  $\mu$ ,  $\{\hat{\rho}_t\}$ ,  $\{\hat{\eta}_t\}$ ,  $\{v_t\}$ , and  $\{u_t\}$ , defined in Algorithm 2 and by the critic recursion (46) converge to  $\rho(\mu)$ ,  $\eta(\mu)$ ,  $v^\mu$ , and  $u^\mu$  with probability one, where  $v^\mu$  and  $u^\mu$  are the unique solutions to

$$\Phi_v^\top \mathbf{D}^\mu \Phi_v v^\mu = \Phi_v^\top \mathbf{D}^\mu T_v^\mu(\Phi_v v^\mu), \quad \Phi_u^\top \mathbf{D}^\mu \Phi_u u^\mu = \Phi_u^\top \mathbf{D}^\mu T_u^\mu(\Phi_u u^\mu), \quad (52)$$

respectively. In (52),  $\mathbf{D}^\mu$  denotes the diagonal matrix with entries  $d^\mu(x)$  for all  $x \in \mathcal{X}$ , and  $T_v^\mu$  and  $T_u^\mu$  are the Bellman operators for the differential value and square value functions of policy  $\mu$ , defined as

$$T_v^\mu J = \mathbf{r}^\mu - \rho(\mu)\mathbf{e} + \mathbf{P}^\mu J, \quad T_u^\mu J = \mathbf{R}^\mu \mathbf{r}^\mu - \eta(\mu)\mathbf{e} + \mathbf{P}^\mu J, \quad (53)$$

where  $\mathbf{r}^\mu$  and  $\mathbf{P}^\mu$  are the reward vector and transition probability matrix of policy  $\mu$ ,  $\mathbf{R}^\mu = \text{diag}(\mathbf{r}^\mu)$ , and  $\mathbf{e}$  is a vector of size  $n$  (the size of the state space  $\mathcal{X}$ ) with elements all equal to one.

*Proof.* The proof follows the same steps as Lemma 5 in [11]. ■

□

## Step 2: Actor's Convergence

**Lemma 10.** *Under (A1)-(A4), given  $\varepsilon > 0$ ,  $\exists \delta > 0$  such that for  $\theta_t$ ,  $t \geq 0$  obtained using Algorithm 2, if  $\sup_{\theta} \|\mathcal{B}(\theta, \lambda)\| < \delta$  then  $\theta_t \rightarrow \mathcal{Z}_{\lambda}^{\varepsilon}$  as  $t \rightarrow \infty$  with probability one.*

*Proof.* See Appendix B. ■

□

## Step 3: $\lambda$ Convergence and Overall Convergence of the Algorithm

As in the discounted setting, we first show that the  $\lambda$ -recursion converges and then prove convergence to a local saddle point of  $L(\theta, \lambda)$ . Consider the ODE

$$\dot{\lambda}_t = \check{\Gamma}_{\lambda}(\Lambda(\theta_t) - \alpha). \quad (54)$$

**Theorem 11.**  $\lambda_t \rightarrow \mathcal{F}$  almost surely as  $t \rightarrow \infty$ , where  $\mathcal{F} \triangleq \{\lambda \mid \lambda \in [0, \lambda_{\max}], \check{\Gamma}_{\lambda}(\Lambda(\theta^{\lambda}) - \alpha) = 0, \theta^{\lambda} \in \mathcal{Z}_{\lambda}\}$ .

*Proof.* The proof follows in a similar manner as that of Theorem 3 in [8]. □

The last step involving the proof of convergence to a (local) saddle point of  $L(\theta, \lambda)$  follows the same steps as that used for the discounted reward algorithms.

## 6.2 Extension of the Algorithm to Sharpe Ratio Optimization

The gradient of the Sharpe Ratio (SR) in the average setting is given by

$$\nabla S(\theta) = \frac{1}{\sqrt{\Lambda(\theta)}} \left( \nabla \rho(\theta) - \frac{\rho(\theta)}{2\Lambda(\theta)} \nabla \Lambda(\theta) \right),$$

and thus, the actor recursion for the SR-variant of our average reward risk-sensitive actor-critic algorithm is as follows:

$$\theta_{t+1} = \Gamma \left( \theta_t + \frac{\zeta_2(t)}{\sqrt{\hat{\eta}_{t+1} - \hat{\rho}_{t+1}^2}} \left( \delta_t \psi_t - \frac{\hat{\rho}_{t+1}(\epsilon_t \psi_t - 2\hat{\rho}_{t+1} \delta_t \psi_t)}{2(\hat{\eta}_{t+1} - \hat{\rho}_{t+1}^2)} \right) \right). \quad (55)$$

Note that the rest of the updates, including the average reward, TD errors, and critic recursions are as in the risk-sensitive actor-critic algorithm presented in Algorithm 2. Similar to the discounted setting, since there is no Lagrange multiplier in the SR optimization, the resulting actor-critic algorithm is a two time-scale stochastic approximation algorithm.

## 7 Experimental Results

We evaluate our algorithms in the context of a traffic signal control application. The objective in our formulation is to minimize the total number of vehicles in the system, which indirectly minimizes the delay experienced by the system. The motivation behind using a risk-sensitive control strategy is to reduce the variations in the delay experienced by road users.

### 7.1 Implementation

We consider both infinite horizon discounted and average settings for the traffic signal control MDP, formulated as in [32]. We briefly recall their formulation here: The state at each time  $t$ ,  $x_t$ , is the vector of queue lengths and elapsed times and is given by  $x_t = (q_1(t), \dots, q_N(t), t_1(t), \dots, t_N(t))$ . Here  $q_i$  and  $t_i$  denote the queue length and elapsed time since the signal turned to red on lane  $i$ . The actions  $a_t$  belong to the set of feasible sign configurations. The single-stage cost function  $h(x_t)$  is defined as follows:

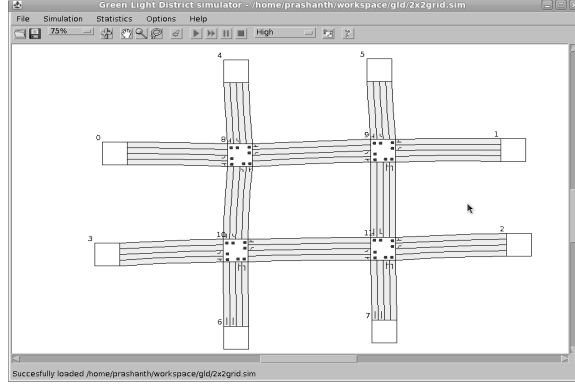


Figure 2: The 2x2-grid network used in our traffic signal control experiments.

$$h(x_t) = r_1 * \left[ \sum_{i \in I_p} r_2 * q_i(t) + \sum_{i \notin I_p} s_2 * q_i(t) \right] + s_1 * \left[ \sum_{i \in I_p} r_2 * t_i(t) + \sum_{i \notin I_p} s_2 * t_i(t) \right], \quad (56)$$

where  $r_i, s_i \geq 0$  such that  $r_i + s_i = 1$  for  $i = 1, 2$  and  $r_2 > s_2$ . The set  $I_p$  is the set of prioritized lanes in the road network considered. While the weights  $r_1, s_1$  are used to differentiate between the queue length and elapsed time factors, the weights  $r_2, s_2$  help in prioritization of traffic.

Given the above traffic control setting, we aim to minimize both the long run discounted and average sum of the cost function  $h(x_t)$ . We implement the following algorithms using the Green Light District (GLD) simulator [49]<sup>5</sup>:

### Discounted Setting

1. **SPSA-G**: This is a first-order risk-neutral algorithm with SPSA-based gradient estimates that updates the parameter  $\theta$  as follows:

$$\theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} + \frac{\zeta_2(t)}{\beta \Delta_t^{(i)}} (v_t^+ - v_t)^\top \phi_v(x^0) \right),$$

where the critic parameters  $v_t, v_t^+$  are updated according to (14). Note that this is a two-timescale algorithm with a TD critic on the faster timescale and the actor on the slower timescale. Unlike RS-SPSA-G, this algorithm, being risk-neutral, does not involve the Lagrange multiplier recursion.

2. **SF-G**: This is a first-order risk-neutral algorithm that is similar to SPSA-G, except that the gradient estimation scheme used here is based on the smoothed functional (SF) technique. The update of the policy parameter in this algorithm is given by

$$\theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} + \zeta_2(t) \left( \frac{\Delta_t^{(i)}}{\beta} (v_t^+ - v_t)^\top \phi_v(x^0) \right) \right).$$

3. **SPSA-N**: This is a risk-neutral algorithm and is the second-order counterpart of SPSA-G. The Hessian update in this algorithm is as follows: For  $i, j = 1, \dots, \kappa_1, i < j$ , the update is

$$H_{t+1}^{(i,j)} = H_t^{(i,j)} + \zeta_2(t) \left[ \frac{(v_t - v_t^+)^\top \phi_v(x^0)}{\beta^2 \Delta_t^{(i)} \hat{\Delta}_t^{(j)}} - H_t^{(i,j)} \right], \quad (57)$$

<sup>5</sup>We would like to point out that the experimental setting involves 'costs' and not 'rewards' and the algorithms implemented should be understood as optimizing a negative reward.

and for  $i > j$ , we set  $H_{t+1}^{(i,j)} = H_{t+1}^{(j,i)}$ . As in RS-SPSA-N, let  $M_t \triangleq H_t^{-1}$ , where  $H_t = \Upsilon([H_t^{(i,j)}]_{i,j=1}^{\kappa_1})$ . The actor updates the parameter  $\theta$  as follows:

$$\theta_{t+1}^{(i)} = \Gamma_i \left[ \theta_t^{(i)} + \zeta_2(t) \sum_{j=1}^{\kappa_1} M_t^{(i,j)} \left( \frac{(v_t^+ - v_t)^\top \phi_v(x^0)}{\beta \Delta_t^{(j)}} \right) \right]. \quad (58)$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.

4. **SF-N**: This is a risk-neutral algorithm and is the second-order counterpart of SF-G. It updates the Hessian and the actor as follows: For  $i, j, k = 1, \dots, \kappa_1$ ,  $j < k$ , the Hessian update is

$$\begin{aligned} \textbf{Hessian:} \quad H_{t+1}^{(i,i)} &= H_t^{(i,i)} + \zeta_2(t) \left[ \frac{(\Delta_t^{(i)})^2 - 1}{\beta^2} (v_t - v_t^+)^\top \phi_v(x^0) - H_t^{(i,i)} \right], \\ H_{t+1}^{(j,k)} &= H_t^{(j,k)} + \zeta_2(t) \left[ \frac{\Delta_t^{(j)} \Delta_t^{(k)}}{\beta^2} (v_t - v_t^+)^\top \phi_v(x^0) - H_t^{(j,k)} \right], \end{aligned}$$

and for  $j > k$ , we set  $H_{t+1}^{(j,k)} = H_{t+1}^{(k,j)}$ . As before, let  $M_t \triangleq H_t^{-1}$ , with  $H_t$  formed as in SPSA-N. Then, the actor update for the parameter  $\theta$  is as follows:

$$\textbf{Actor:} \quad \theta_{t+1}^{(i)} = \Gamma_i \left[ \theta_t^{(i)} + \zeta_2(t) \sum_{j=1}^{\kappa_1} M_t^{(i,j)} \frac{\Delta_t^{(j)}}{\beta} (v_t^+ - v_t)^\top \phi_v(x^0) \right].$$

The rest of the symbols, including the critic parameters, are as in RS-SPSA-N.

5. **RS-SPSA-G**: This is the first-order risk-sensitive actor-critic algorithm that attempts to solve (40) and updates according to (18).
6. **RS-SF-G**: This is a first-order algorithm and the risk-sensitive variant of SF-G that updates the actor according to (19).
7. **RS-SPSA-N**: This is a second-order risk-sensitive algorithm that estimates gradient and Hessian using SPSA and updates them according to (24).
8. **RS-SF-N**: This second-order risk-sensitive algorithm is the SF counterpart of RS-SPSA-N, and updates according to (29).

### Average Setting

1. **AC**: This is an actor-critic algorithm that minimizes the long-run average sum of the single-stage cost function  $h(x_t)$ , without considering any risk criteria. This is similar to Algorithm 1 in Bhatnagar et al. [11].
2. **RS-AC**: This is the risk-sensitive actor-critic algorithm that attempts to solve (40) and is described in Section 6.

The underlying policy that guides the selection of the sign configuration in each of the algorithms above is a parameterized Boltzmann family and has the form

$$\mu_\theta(x, a) = \frac{e^{\theta^\top \phi_{x,a}}}{\sum_{a' \in \mathcal{A}(x)} e^{\theta^\top \phi_{x,a'}}}, \quad \forall x \in \mathcal{X}, \forall a \in \mathcal{A}. \quad (59)$$

All our algorithms incorporate function approximation owing to the curse of dimensionality associated with larger road networks. For instance, assuming only 20 vehicles per lane of a 2x2-grid network, the cardinality of the state space is approximately of the order  $10^{32}$  and the situation is aggravated as the size of the road network increases. The choice of features used in each of our algorithms is as described in Section V-B of [33].

The experiments for each algorithm comprised of the following two phases:

**Policy Search Phase:** Here each iteration involved the simulation run with the nominal policy parameter  $\theta$  as well as the perturbed policy parameter  $\theta^+$  (algorithm-specific). We run each algorithm for 500 iterations, where the run length for a particular policy parameter is 150 steps.

**Policy Test Phase:** After the completion of the policy search phase, we freeze the policy parameter and run 50 independent simulations with this (converged) choice of the parameter. The results presented subsequently are averages over these 50 runs.

Figure 2 shows a snapshot of the road network used for conducting the experiments from GLD simulator. Traffic is added to the network at each time step from the edge nodes. The spawn frequencies specify the rate at which traffic is generated at each edge node and follow a Poisson distribution. The spawn frequencies are set such that the proportion of the number of vehicles on the main roads (the horizontal ones in Fig. 2) to those on the side roads is in the ratio of 100 : 5. This setting is close to what is observed in practice and has also been used for instance in [32, 33]. In all our experiments, we set the weights in the single stage cost function (56) as follows:  $r_1 = r_2 = 0.5$  and  $r_2 = 0.6, s_2 = 0.4$ . For the SPSA and SF-based algorithms in the discounted setting, we set the parameter  $\delta = 0.2$  and the discount factor  $\gamma = 0.9$ . The parameter  $\alpha$  in the formulations (40) and (3) was set to 20. The step-size sequences are chosen as follows:

$$\zeta_1(t) = \frac{1}{t}, \quad \zeta_2(t) = \frac{1}{t^{0.75}}, \quad \zeta_3(t) = \frac{1}{t^{0.66}}, \quad t \geq 1. \quad (60)$$

Further, the constant  $k$  related to  $\zeta_4(t)$  in the risk-sensitive average reward algorithm is set to 1. It is easy to see that the choice of step-sizes above satisfies (A4). The projection operator  $\Gamma_i$  was set to project the iterate  $\theta^{(i)}$  onto the set  $[0, 10]$ , for all  $i = 1, \dots, \kappa_1$ , while the projection operator for the Lagrange multiplier used the set  $[0, 1000]$ . All the experiments were performed on a 2.53GHz Intel quad core machine with 3.8GB RAM.

## 7.2 Results

Figure 3 shows the distribution of the discounted cumulative reward  $D^\theta(x^0)$  for the algorithms in the discounted setting. Figure 4 shows the total arrived road users (TAR) obtained for all the algorithms in the discounted setting, whereas Figure 5 presents the average junction waiting time (AJWT) for the first-order SF-based algorithm RS-SF-G.<sup>6</sup> TAR is a throughput metric that measures the number of road users who have reached their destination, whereas AJWT is a delay metric that quantifies the average delay experienced by the road users.

The performance of the algorithms in the average setting is presented in Figure 6. In particular, Figure 6(a) shows the distribution of the average reward  $\rho$ , while Figure 6(b) presents the average junction waiting time (AJWT) for the average cost algorithms.

From Figures 3 and 6(a), we notice that the risk-sensitive algorithms proposed in this paper result in a long-term (discounted or average) cost that is higher than their risk-neutral variants. However, from the empirical variance of the cost (both discounted as well as average) perspective, the risk-sensitive algorithms outperform their risk-neutral variants. Amongst our algorithms in the discounted setting, we observe that the second-order schemes (RS-SPSA-N and RS-SF-N) exhibit better results, though they involve an additional computational cost of inverting the Hessian at each time step. Further, from a traffic signal control application standpoint, we notice from the throughput (TAR) and delay (AJWT) plots (see Figures 4, 5 and 6(b)), that the performance of the risk-sensitive algorithm variants is close to that of the corresponding risk-neutral algorithms in both the considered settings.

We observe that the policy parameter  $\theta$  converges for the SPSA based algorithms in the discounted setting. This is illustrated in Figures 7(a) and 7(b). Note that we established theoretical convergence of our algorithms earlier (see Sections 4.5 and 6.1) and these plots confirm the same. Further, these plots also show that the transient period, i.e., the initial phase when  $\theta$  has not converged, is short. Similar observations hold for the other algorithms as well. The results of this section indicate the rapid empirical convergence of our proposed algorithms. This observation coupled with the fact that they guarantee low variance of return, make them attractive for implementation in risk-constrained systems.

<sup>6</sup>The AJWT performance of the other algorithms in the discounted setting is similar and the corresponding plots are omitted here.



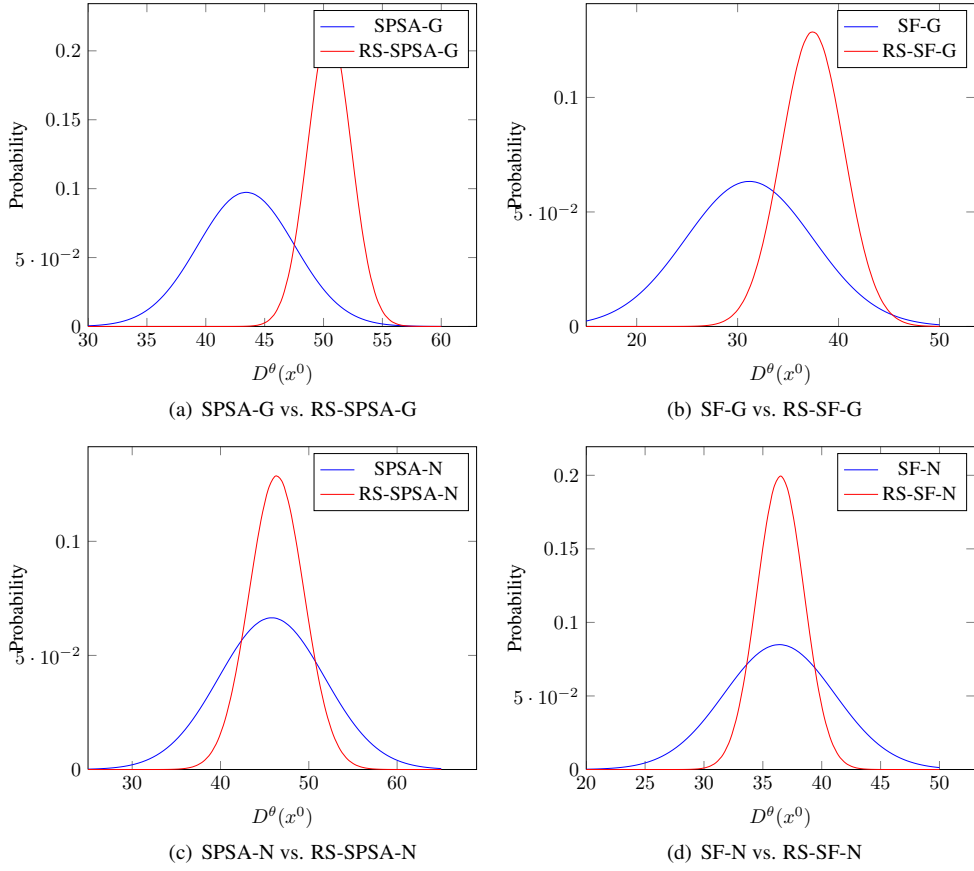
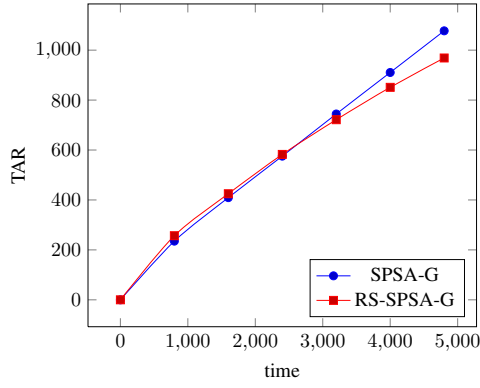
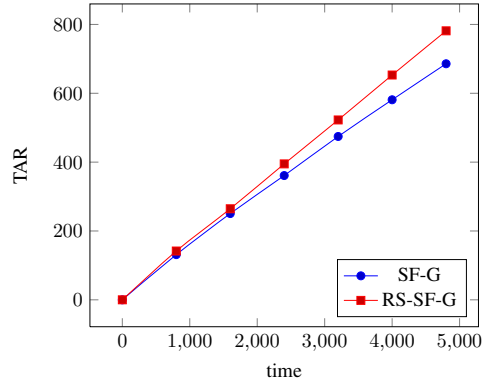


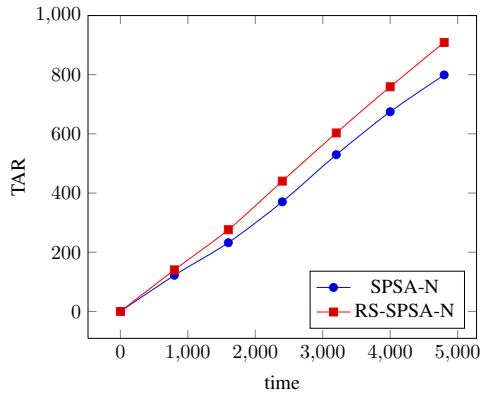
Figure 3: Performance comparison in the discounted setting using the distribution of  $D^\theta(x^0)$ .



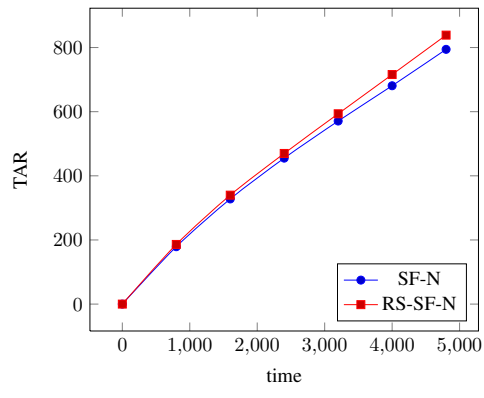
(a) SPSA-G vs. RS-SPSA-G



(b) SF-G vs. RS-SF-G



(c) SPSA-N vs. RS-SPSA-N



(d) SF-N vs. RS-SF-N

Figure 4: Performance comparison of the algorithms in the discounted setting using the total arrived road users (TAR).

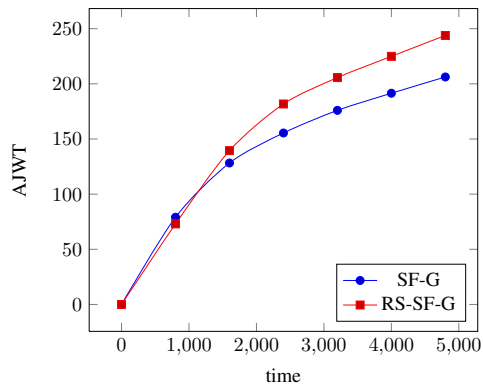
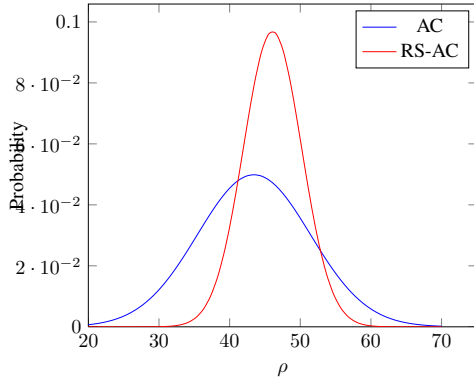
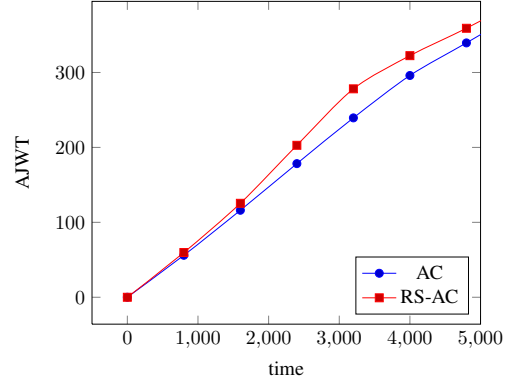


Figure 5: Performance comparison of the first-order SF-based algorithms, SF-G and RS-SF-G, using the average junction waiting time (AJWT).

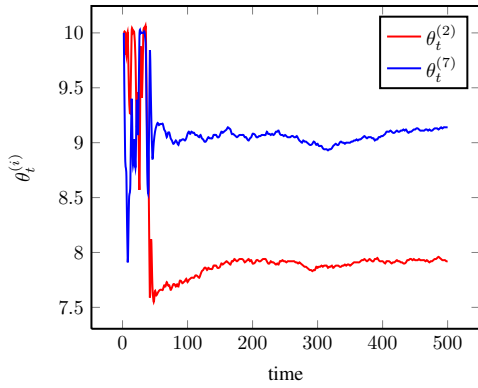


(a) average reward  $\rho$  distribution

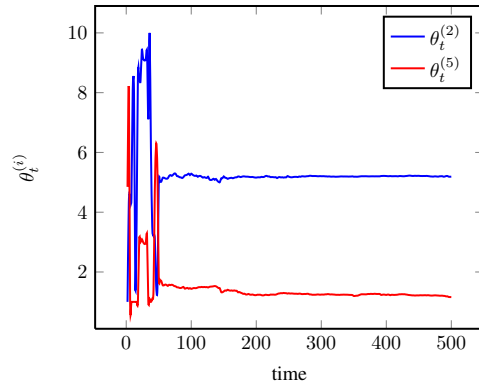


(b) average junction waiting time

Figure 6: Performance comparison of the risk-neutral (AC) and risk-sensitive (RS-AC) average reward actor-critic algorithms using two different metrics.



(a) RS-SPSA-G



(b) RS-SPSA-N

Figure 7: Convergence of SPSA based algorithms in the discounted setting – illustration using two (arbitrarily chosen) coordinates of the parameter  $\theta$ .

## 8 Conclusions and Future Work

We proposed novel actor-critic algorithms for control in risk-sensitive discounted and average reward MDPs. All our algorithms involve a TD critic on the fast timescale, a policy gradient (actor) on the intermediate timescale, and a dual ascent for Lagrange multipliers on the slowest timescale. In the discounted setting, we pointed out the difficulty in estimating the gradient of the variance of the return and incorporated simultaneous perturbation based SPSA and SF approaches for gradient estimation in our algorithms. The average setting, on the other hand, allowed for an actor to employ compatible features to estimate the gradient of the variance. We provided proofs of convergence to locally (risk-sensitive) optimal policies for all the proposed algorithms. Further, using a traffic signal control application, we observed that our algorithms resulted in lower variance empirically as compared to their risk-neutral counterparts.

As future work, it would be interesting to develop a risk-sensitive algorithm that uses a single trajectory in the discounted setting. Further, it would also be interesting to consider conditional value at risk (CVaR) as a measure of risk and develop a control algorithm that optimizes the return of a MDP with bounds on CVaR. The resulting algorithm could be applied for portfolio optimization in a financial application. An orthogonal direction of future research is to obtain finite-time bounds on the quality of the solution obtained by our algorithms. As mentioned earlier, this is challenging as, to the best of our knowledge, there are no convergence rate results available for multi-timescale stochastic approximation schemes, and hence, for actor-critic algorithms.

## References

- [1] A. Barto, R. Sutton, and C. Anderson. Neuron-like elements that can solve difficult learning control problems. *IEEE Transaction on Systems, Man and Cybernetics*, 13:835–846, 1983.
- [2] J. Baxter and P. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.
- [3] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.
- [4] D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.
- [5] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.
- [6] S. Bhatnagar. Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. *ACM Transactions on Modeling and Computer Simulation*, 15(1):74–107, 2005.
- [7] S. Bhatnagar. Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization. *ACM Transactions on Modeling and Computer Simulation*, 18(1):1–35, 2007.
- [8] S. Bhatnagar and K. Lakshmanan. An online actor-critic algorithm with function approximation for constrained Markov decision processes. *Journal of Optimization Theory and Applications*, pages 1–21, 2012.
- [9] S. Bhatnagar, M.C. Fu, S.I. Marcus, and I. Wang. Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation*, 13(2):180–209, 2003. ISSN 1049-3301.
- [10] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Incremental natural actor-Critic algorithms. In *Proceedings of Advances in Neural Information Processing Systems 20*, pages 105–112, 2007.
- [11] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11): 2471–2482, 2009.
- [12] S. Bhatnagar, N. Hemachandra, and V. Mishra. Stochastic approximation algorithms for constrained optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, 21(3):15, 2011.

- [13] S. Bhatnagar, H. Prasad, and L.A. Prashanth. *Stochastic Recursive Algorithms for Optimization*, volume 434. Springer, 2013.
- [14] V. Borkar. A sensitivity formula for the risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44:339–346, 2001.
- [15] V. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27:294–311, 2002.
- [16] V. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems & Control Letters*, 54(3):207–213, 2005.
- [17] V. Borkar. *Stochastic approximation: a dynamical systems viewpoint*. Cambridge University Press, 2008.
- [18] V. Borkar. Learning algorithms for risk-sensitive control. In *Proceedings of the Nineteenth International Symposium on Mathematical Theory of Networks and Systems*, 2010.
- [19] H. Chen, T. Duncan, and B. Pasik-Duncan. A Kiefer-Wolfowitz algorithm with randomized differences. *IEEE Transactions on Automatic Control*, 44(3):442–453, 1999.
- [20] E. Delage and S. Mannor. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58(1):203–213, 2010.
- [21] J. Filar, L. Kallenberg, and H. Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.
- [22] J. Filar, D. Krass, and K. Ross. Percentile performance criteria for limiting average Markov decision processes. *IEEE Transaction of Automatic Control*, 40(1):2–10, 1995.
- [23] R. Howard and J. Matheson. Risk sensitive Markov decision processes. *Management Science*, 18(7):356–369, 1972.
- [24] V. Katkovnik and Y. Kulchitsky. Convergence of a class of random search algorithms. *Automatic Remote Control*, 8:81–87, 1972.
- [25] V. Konda and J. Tsitsiklis. Actor-Critic algorithms. In *Proceedings of Advances in Neural Information Processing Systems 12*, pages 1008–1014, 2000.
- [26] H. Kushner and D. Clark. *Stochastic approximation methods for constrained and unconstrained systems*. Springer-Verlag, 1978.
- [27] S. Mannor and J. Tsitsiklis. Mean-variance optimization in Markov decision processes. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, pages 177–184, 2011.
- [28] P. Marbach. *Simulated-Based Methods for Markov Decision Processes*. PhD thesis, Massachusetts Institute of Technology, 1998.
- [29] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic theory*. Oxford University Press, 1995.
- [30] A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.
- [31] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the Sixteenth European Conference on Machine Learning*, pages 280–291, 2005.
- [32] L.A. Prashanth and S. Bhatnagar. Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):412–421, june 2011.
- [33] L.A. Prashanth and S. Bhatnagar. Threshold Tuning Using Stochastic Optimization for Graded Signal Control. *IEEE Transactions on Vehicular Technology*, 61(9):3865–3880, nov. 2012.

- [34] L.A. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Proceedings of Advances in Neural Information Processing Systems 26*, pages 252–260, 2013.
- [35] M. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 1994.
- [36] W. Sharpe. Mutual fund performance. *Journal of Business*, 39(1):119–138, 1966.
- [37] M. Sobel. The variance of discounted Markov decision processes. *Applied Probability*, pages 794–802, 1982.
- [38] J. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.
- [39] J. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1):109–112, 1997. ISSN 0005-1098.
- [40] J. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45(10):1839–1853, 2000.
- [41] M. A. Styblinski and L. J. Opalski. Algorithms and software tools for IC yield optimization based on fundamental fabrication parameters. *IEEE Transactions on Computer Aided Design CAD*, 1(5):79–89, 1986.
- [42] R. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 1984.
- [43] R. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.
- [44] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998.
- [45] R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of Advances in Neural Information Processing Systems 12*, pages 1057–1063, 2000.
- [46] A. Tamar and S. Mannor. Variance adjusted actor-critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.
- [47] A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the Twenty-Ninth International Conference on Machine Learning*, pages 387–396, 2012.
- [48] A. Tamar, D. Di Castro, and S. Mannor. Temporal difference methods for the variance of the reward to go. In *Proceedings of the Thirtieth International Conference on Machine Learning*, pages 495–503, 2013.
- [49] M. Wiering, J. Vreeken, J. van Veenen, and A. Koopman. Simulation and optimization of traffic in a city. In *IEEE Intelligent Vehicles Symposium*, pages 453–458, June 2004.
- [50] R. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.
- [51] H. Xu and S. Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

## A Appendix: Discounted Reward Proofs

### Proof of Theorem 2

*Proof.* The proof of this theorem follows similar steps as in the proof of Theorem 10 in Tamar et al. [48]. For our analysis, we need to extend their proof to discounted MDPs and to the case that the reward is a function of both states and actions (and not just states), which is straightforward.  $\blacksquare$

### Proof of Theorem 3 for RS-SPSA-G

*Proof.* Owing to timescale separation between  $\theta$  and  $\lambda$  recursions in RS-SPSA-G, we treat  $\lambda_t \equiv \lambda$ , a constant in the analysis of  $\theta$ -recursion (18). Further, since the TD critic converges on the faster timescale, the  $\theta$ -update in (18) can be rewritten using the converged TD-parameters  $(\bar{v}, \bar{u})$  and  $(\bar{v}^+, \bar{u}^+)$  as

$$\theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} - \zeta_2(t) \left( - (1 + 2\lambda \bar{v}^\top \phi_v(x^0)) \frac{(\bar{v}^+ - \bar{v})^\top \phi_v(x^0)}{\beta \Delta_t^{(i)}} + \lambda \frac{(\bar{u}^+ - \bar{u})^\top \phi_u(x^0)}{\beta \Delta_t^{(i)}} + \xi_{1,t} \right) \right),$$

where  $\xi_{1,t} \rightarrow 0$  (convergence of TD in the critic and as a result convergence of the critic's parameters to  $\bar{v}, \bar{u}, \bar{v}^+, \bar{u}^+$ ) in lieu of Theorem 2.

Next, we establish that  $\mathbb{E} \left[ \frac{(\bar{v}^+ - \bar{v})^\top \phi_v(x^0)}{\beta \Delta^{(i)}} \mid \theta, \lambda \right]$  is a biased estimator of  $\nabla_\theta \hat{V}(\theta)$ , where the bias vanishes asymptotically, i.e.,

$$\begin{aligned} & \mathbb{E} \left[ \frac{(\bar{v}^+ - \bar{v})^\top \phi_v(x^0)}{\beta \Delta^{(i)}} \mid \theta, \lambda \right] \nabla_i \bar{v}^\top \phi_v(x^0) + \mathbb{E} \left[ \sum_{j \neq i} \frac{\Delta^{(j)}}{\Delta^{(i)}} \nabla_j \bar{v}^\top \phi_v(x^0) \mid \theta, \lambda \right] + \xi_{2,t} \phi_v(x^0) \\ & \xrightarrow{\beta \rightarrow 0} \nabla_i \bar{v}^\top \phi_v(x^0). \end{aligned}$$

The first equality above follows by expanding using Taylor series, whereas the second step follows by using the fact that  $\Delta_t^{(i)}$ 's are independent Rademacher random variables. On similar lines, it can be seen that

$$\mathbb{E} \left[ \frac{(\bar{u}^+ - \bar{u})^\top \phi_u(x^0)}{\beta \Delta^{(i)}} \mid \theta, \lambda \right] \xrightarrow{\beta \rightarrow 0} \nabla_i \bar{u}^\top \phi_u(x^0).$$

Thus, (18) can be seen to be a discretization of the ODE (30). Further,  $\mathcal{Z}_\lambda$  is an asymptotically stable attractor for the ODE (30), with  $\hat{L}(\theta, \lambda)$  itself serving as a strict Lyapunov function. This can be inferred as follows:

$$\frac{d\hat{L}(\theta, \lambda)}{dt} = \nabla_\theta \hat{L}(\theta, \lambda) \dot{\theta} = \nabla_\theta \hat{L}(\theta, \lambda) \check{\Gamma}(-\nabla_\theta \hat{L}(\theta, \lambda)) < 0.$$

The claim now follows from Theorem 5.3.3, pp. 191-196 of Kushner and Clark [26].  $\blacksquare$

### Proof of Theorem 3 for RS-SF-G

*Proof.* As in the case of the SPSA algorithm, we rewrite the  $\theta$ -update in (19) using the converged TD-parameters and constant  $\lambda$  as

$$\begin{aligned} \theta_{t+1}^{(i)} = \Gamma_i \left( \theta_t^{(i)} - \zeta_2(t) \left( \frac{-\Delta_t^{(i)} (1 + 2\lambda \bar{v}^\top \phi_v(x^0))}{\beta} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) \right. \right. \\ \left. \left. + \frac{\lambda \Delta_t^{(i)}}{\beta} (\bar{u}^+ - \bar{u})^\top \phi_u(x^0) + \xi_{1,t} \right) \right), \end{aligned}$$

where  $\xi_{1,t} \rightarrow 0$  (convergence of TD in the critic and as a result convergence of the critic's parameters to  $\bar{v}, \bar{u}, \bar{v}^+, \bar{u}^+$ ) in lieu of Theorem 2. Next, we establish that

$\mathbb{E} \left[ \frac{\Delta^{(i)}}{\beta} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) \mid \theta, \lambda \right]$  is an asymptotically correct estimate of the gradient of  $\widehat{V}(\theta)$  in the following:

$$\mathbb{E} \left[ \frac{\Delta^{(i)}}{\beta} (\bar{v}^+ - \bar{v})^\top \phi_v(x^0) \mid \theta, \lambda \right] \xrightarrow{\beta \rightarrow 0} \nabla_i \bar{v}^\top \phi_v(x^0).$$

The above follows in a similar manner as Proposition 10.2 of Bhatnagar et al. [13]. On similar lines, one can see that

$$\mathbb{E} \left[ \frac{\Delta^{(i)}}{\beta} (\bar{u}^+ - \bar{u})^\top \phi_u(x^0) \mid \theta, \lambda \right] \xrightarrow{\beta \rightarrow 0} \nabla_i \bar{u}^\top \phi_u(x^0).$$

Thus, (19) can be seen to be a discretization of the ODE (30) and the rest of the analysis follows in a similar manner as in the SPSA proof.  $\blacksquare$

$\square$

### **Proof of Theorem 5 for RS-SPSA-N**

Before we prove Theorem 5, we establish that the Hessian estimate  $H_t$  in (23) converges almost surely to the true Hessian  $\nabla_\theta^2 L(\theta_t, \lambda)$  in the following lemma.

**Lemma 12.** *With  $\beta \rightarrow 0$  as  $t \rightarrow \infty$ , for all  $i, j \in \{1, \dots, \kappa_1\}$ , we have the following claims with probability one:*

$$(i) \left\| \frac{L(\theta_t + \beta \Delta_t + \beta \widehat{\Delta}_t, \lambda) - L(\theta_t, \lambda)}{\beta^2 \Delta_t^{(i)} \widehat{\Delta}_t^{(j)}} - \nabla_{\theta^{(i,j)}}^2 L(\theta_t, \lambda) \right\| \rightarrow 0,$$

$$(ii) \left\| \frac{L(\theta_t + \beta \Delta_t + \beta \widehat{\Delta}_t, \lambda) - L(\theta_t, \lambda)}{\beta \widehat{\Delta}_t^{(i)}} - \nabla_{\theta^{(i)}} L(\theta_t, \lambda) \right\| \rightarrow 0,$$

$$(iii) \left\| H_t^{(i,j)} - \nabla_{\theta^{(i,j)}}^2 L(\theta_t, \lambda) \right\| \rightarrow 0,$$

$$(iv) \left\| M_t - \Upsilon(\nabla_\theta^2 L(\theta_t, \lambda))^{-1} \right\| \rightarrow 0.$$

*Proof.* The proofs of the above claims follow from Propositions 10.10, 10.11 and Lemmas 7.10 and 7.11 of [13], respectively.  $\blacksquare$

$\square$

*Proof. (Theorem 5 for RS-SPSA-N)* As in the case of the first order methods, due to timescale separation, we can treat  $\lambda_t \equiv \lambda$ , a constant and use the converged TD-parameters to arrive at the following equivalent update rules for the Hessian recursion (23) and  $\theta$ -recursion (24):

$$\begin{aligned} H_{t+1}^{(i,j)} &= H_t^{(i,j)} + \zeta_2(t) \left[ \frac{(1 + \lambda_t (\bar{v}_t + \bar{v}_t^+)^\top \phi_v(x^0)) (\bar{v}_t - \bar{v}_t^+)^\top \phi_v(x^0)}{\beta^2 \Delta_t^{(i)} \widehat{\Delta}_t^{(j)}} \right. \\ &\quad \left. + \frac{\lambda (\bar{u}_t^+ - \bar{u}_t)^\top \phi_u(x^0)}{\beta^2 \Delta_t^{(i)} \widehat{\Delta}_t^{(j)}} - H_t^{(i,j)} \right], \\ \theta_{t+1}^{(i)} &= \Gamma_i \left[ \theta_t^{(i)} + \zeta_2(t) \sum_{j=1}^{\kappa_1} M_t^{(i,j)} \left( \frac{(1 + 2\lambda \bar{v}_t^\top \phi_v(x^0)) (\bar{v}_t^+ - \bar{v}_t)^\top \phi_v(x^0)}{\beta \Delta_t^{(j)}} \right. \right. \\ &\quad \left. \left. - \frac{\lambda (\bar{u}_t^+ - \bar{u}_t)^\top \phi_u(x^0)}{\beta \Delta_t^{(j)}} \right) \right]. \end{aligned}$$



In lieu of Lemma 12, the  $\theta$ -recursion above is equivalent to the following:

$$\theta_{t+1}^{(i)} = \bar{\Gamma}_i \left( \theta_t^{(i)} + \zeta_2(t) (\nabla_{\theta}^2 L(\theta_t, \lambda))^{-1} \nabla_{\theta} L(\theta_t, \lambda) \right). \quad (61)$$

The above can be seen as a discretization of the ODE (34), with  $\mathcal{Z}_{\lambda}$  serving as its asymptotically stable attractor. The rest of the claim follows in a similar manner as Theorem 3.  $\blacksquare$   $\square$

### **Proof of Theorem 5 for RS-SF-N**

*Proof.* We first establish the following result for the gradient and Hessian estimators employed in RS-SF-N:

**Lemma 13.** *With  $\beta \rightarrow 0$  as  $t \rightarrow \infty$ , we have the following claims with probability one:*

$$(i) \left\| E \left[ \frac{1}{\beta^2} \bar{H}(\Delta_t) (L(\theta_t + \beta \Delta_t, \lambda) - L(\theta_t, \lambda)) \mid \theta_t, \lambda \right] - \nabla_{\theta}^2 L(\theta_t, \lambda) \right\| \rightarrow 0.$$

$$(ii) \left\| E \left[ \frac{1}{\beta} \Delta_t (L(\theta_t + \beta \Delta_t, \lambda) - L(\theta_t, \lambda)) \mid \theta_t, \lambda \right] - \nabla_{\theta} L(\theta_t, \lambda) \right\| \rightarrow 0.$$

*Proof.* The proofs of the above claims follow from Propositions 10.1 and 10.2 of [13], respectively.  $\blacksquare$   $\square$

The rest of the analysis is identical to that of RS-SPSA-N.  $\blacksquare$   $\square$

## **B Appendix: Average Reward Proofs**

### **Proof of Lemma 8**

*Proof.* The bias in estimating  $\nabla L(\theta, \lambda)$  consists of the bias in estimating  $\nabla \rho(\theta)$  and  $\nabla \eta(\theta)$ . Lemma 4 in Bhatnagar et al. [11] shows the bias in estimating  $\nabla \rho(\theta)$  as

$$\mathbb{E}[\delta_t^{\theta} \psi_t | \theta] = \nabla \rho(\theta) + \sum_{x \in \mathcal{X}} d^{\theta}(x) [\nabla \bar{V}^{\theta}(x) - \nabla v^{\theta \top} \phi_v(x)],$$

where  $\delta_t^{\theta} = R(x_t, a_t) - \hat{\rho}_{t+1} + v^{\theta \top} \phi_v(x_{t+1}) - v^{\theta \top} \phi_v(x_t)$ . Similarly we can prove that the bias in estimating  $\nabla \eta(\theta)$  is

$$\mathbb{E}[\epsilon_t^{\theta} \psi_t | \theta] = \nabla \eta(\theta) + \sum_{x \in \mathcal{X}} d^{\theta}(x) [\nabla \bar{U}^{\theta}(x) - \nabla u^{\theta \top} \phi_u(x)],$$

where  $\epsilon_t^{\theta} = R(x_t, a_t) - \hat{\eta}_{t+1} + u^{\theta \top} \phi_u(x_{t+1}) - u^{\theta \top} \phi_u(x_t)$ . The claim follows by putting these two results together and given the fact that  $\nabla \Lambda(\theta) = \nabla \eta(\theta) - 2\rho(\theta) \nabla \rho(\theta)$  and  $\nabla L(\theta, \lambda) = -\nabla \rho(\theta) + \lambda \nabla \Lambda(\theta)$ . Note that the following fact holds for the bias in estimating  $\nabla \rho(\theta)$  and  $\nabla \eta(\theta)$ :

$$\sum_x d^{\theta}(x) [\bar{V}^{\theta}(x) - v^{\theta \top} \phi_v(x)] = 0, \quad \sum_x d^{\theta}(x) [\bar{U}^{\theta}(x) - u^{\theta \top} \phi_u(x)] = 0. \quad \blacksquare$$

$\square$

### **Proof of Lemma 10**

*Proof.* First note that the bias of Algorithm 2 in estimating  $\nabla L(\theta, \lambda)$  is (see Lemma 8)

$$\mathcal{B}(\theta, \lambda) = \sum_x d^\theta(x) \left\{ - (1 + 2\lambda\rho(\theta)) [\nabla \bar{V}^\theta(x) - \nabla v^{\theta^\top} \phi_v(x)] + \lambda [\nabla \bar{U}^\theta(x) - \nabla u^{\theta^\top} \phi_u(x)] \right\}.$$

Also note that  $\mathcal{Z}_\lambda = \{\theta \in C : \tilde{\Gamma}(-\nabla L(\theta, \lambda)) = 0\}$  denote the set of asymptotically stable equilibrium points of the ODE (30) and  $\mathcal{Z}_\lambda^\varepsilon = \{\theta \in C : \|\theta - \theta_0\| < \varepsilon, \theta_0 \in \mathcal{Z}_\lambda\}$  denote the set of points in the  $\varepsilon$ -neighborhood of  $\mathcal{Z}_\lambda$ . Let  $\mathcal{F}(t) = \sigma(\theta_r, r \leq t)$  denote the sequence of  $\sigma$ -fields generated by  $\theta_r$ ,  $r \geq 0$ . We have

$$\begin{aligned} \theta_{t+1} &= \Gamma\left(\theta_t - \zeta_2(t) (-\delta_t \psi_t + \lambda(\epsilon_t \psi_t - 2\hat{\rho}_{t+1} \delta_t \psi_t))\right) \\ &= \Gamma\left(\theta_t + \zeta_2(t)(1 + 2\lambda\hat{\rho}_{t+1})\delta_t \psi_t - \zeta_2(t)\lambda\epsilon_t \psi_t\right) \\ &= \Gamma\left(\theta_t - \zeta_2(t) \left[1 + 2\lambda\left(\hat{\rho}_{t+1} - \rho(\theta_t)\right) + \rho(\theta_t)\right] \mathbb{E}[\delta^{\theta_t} \psi_t | \mathcal{F}(t)] \right. \\ &\quad \left. - \zeta_2(t) \left[1 + 2\lambda\left(\hat{\rho}_{t+1} - \rho(\theta_t)\right) + \rho(\theta_t)\right] \left(\delta_t \psi_t - \mathbb{E}[\delta_t \psi_t | \mathcal{F}(t)]\right) \right. \\ &\quad \left. - \zeta_2(t) \left[1 + 2\lambda\left(\hat{\rho}_{t+1} - \rho(\theta_t)\right) + \rho(\theta_t)\right] \mathbb{E}[(\delta_t - \delta^{\theta_t}) \psi_t | \mathcal{F}(t)] \right. \\ &\quad \left. + \zeta_2(t)\lambda \mathbb{E}[\epsilon^{\theta_t} \psi_t | \mathcal{F}(t)] + \zeta_2(t)\lambda \left(\epsilon_t \psi_t - \mathbb{E}[\epsilon_t \psi_t | \mathcal{F}(t)]\right) \right. \\ &\quad \left. + \zeta_2(t)\lambda \mathbb{E}[(\epsilon_t - \epsilon^{\theta_t}) \psi_t | \mathcal{F}(t)]\right). \end{aligned}$$

By setting  $\xi_t = \hat{\rho}_{t+1} - \rho(\theta_t)$ , we may write the above equation as

$$\theta_{t+1} = \Gamma\left(\theta_t - \zeta_2(t) \left[1 + 2\lambda(\xi_t + \rho(\theta_t))\right] \mathbb{E}[\delta^{\theta_t} \psi_t | \mathcal{F}(t)] \right. \quad (62)$$

$$\begin{aligned} &\quad \left. - \zeta_2(t) \left[1 + 2\lambda(\xi_t + \rho(\theta_t))\right] \underbrace{\left(\delta_t \psi_t - \mathbb{E}[\delta_t \psi_t | \mathcal{F}(t)]\right)}_{*} \right. \\ &\quad \left. - \zeta_2(t) \left[1 + 2\lambda(\xi_t + \rho(\theta_t))\right] \underbrace{\mathbb{E}[(\delta_t - \delta^{\theta_t}) \psi_t | \mathcal{F}(t)]}_{+} \right. \\ &\quad \left. + \zeta_2(t)\lambda \mathbb{E}[\epsilon^{\theta_t} \psi_t | \mathcal{F}(t)] + \zeta_2(t)\lambda \underbrace{\left(\epsilon_t \psi_t - \mathbb{E}[\epsilon_t \psi_t | \mathcal{F}(t)]\right)}_{*} \right. \\ &\quad \left. + \zeta_2(t)\lambda \underbrace{\mathbb{E}[(\epsilon_t - \epsilon^{\theta_t}) \psi_t | \mathcal{F}(t)]}_{+} \right). \end{aligned} \quad (63)$$

Since Algorithm 2 uses an unbiased estimator for  $\rho$ , we have  $\hat{\rho}_{t+1} \rightarrow \rho(\theta_t)$ , and thus,  $\xi_t \rightarrow 0$ . The terms (+) asymptotically vanish in lieu of Lemma 9 (Critic convergence). Finally the terms (\*) can be seen to vanish using standard martingale arguments (cf. Theorem 2 in [11]). Thus, (62) can be seen to be equivalent in an asymptotic sense to

$$\theta_{t+1} = \Gamma\left(\theta_t - \zeta_2(t) \left[1 + 2\lambda\rho(\theta_t)\right] \mathbb{E}[\delta^{\theta_t} \psi_t | \mathcal{F}(t)] + \zeta_2(t)\lambda \mathbb{E}[\epsilon^{\theta_t} \psi_t | \mathcal{F}(t)]\right). \quad (64)$$

From Lemma 8 and the foregoing, (47) asymptotically tracks the stable fixed points of the ODE

$$\dot{\theta}_t = \tilde{\Gamma}\left(\nabla L(\theta_t, \lambda) + \mathcal{B}(\theta_t, \lambda)\right). \quad (65)$$

So, if the bias  $\sup_\theta \|\mathcal{B}(\theta, \lambda)\| \rightarrow 0$ , the trajectories (65) converge to those of (30) uniformly on compacts for the same initial condition and the claim follows.  $\blacksquare$   $\square$